

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Tadas Vizgaitis

**MagicDraw UML įskiepio veiklos taisyklėms
modeliuoti sukūrimas ir realizavimas**

Magistro darbas

Darbo vadovas

lekt., dr. K.Kapočius

Kaunas, 2009

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Tadas Vizgaitis

**MagicDraw UML įskiepio veiklos taisyklėms
modeliuoti sukūrimas ir realizavimas**

Magistro darbas

Recenzentas

dr. D. Šilingas
2009-01-12

Vadovas

lekt., dr. K.Kapočius
2009-01-12

Atliko

IFM-3/4 gr. stud.
Tadas Vizgaitis
2009-01-12

Kaunas, 2009

Turinys

SUTRUMPINIMŲ ŽODYNAS	4
1. ĮVADAS	5
2. KOMPIUTERIZUOTO VEIKLOS TAISYKLIŲ MODELIAVIMO IS PROJEKTAVIMO STADIJOJE ANALIZĖ	7
2.1. VEIKLOS TAISYKLIŲ VAIDMUO IS KŪRIMO PROCESE IR TYRIMO KONCEPCIJA	7
2.2. AKTUALIŲ SPRENDIMŲ ANALIZĖ	8
2.2.1. <i>Grafinis veiklos taisyklių modeliavimas: Roso metodo realizacijos prototipas</i>	10
2.3. ROSO METODAS	11
2.3.1. <i>Metodo koncepcija ir sudėtis</i>	11
2.3.2. <i>Veiklos taisyklių formalizavimas pagal Roso metodą</i>	15
2.3.3. <i>Formalūs reikalavimai veiklos taisyklėms</i>	17
2.3.4. <i>Pagal Roso metodą formalizuotų veiklos taisyklių saugyklos modelis</i>	17
2.3.5. <i>Veiklos taisyklių struktūrizavimo eiga</i>	19
2.4. VEIKLOS TAISYKLIŲ MODELIAVIMO ĮRANKIO VARTOTOJŲ ANALIZĖ	20
2.5. MAGICDRAW UML PRODUKTO PRAPLĖTIMO GALIMYBIŲ ANALIZĖ	22
2.5.1. <i>MagicDraw UML profilio kūrimas</i>	22
2.5.2. <i>MagicDraw įskiepis</i>	24
3. MAGICDRAW UML ĮSKIEPIO ROSO VEIKLOS TAISYKLĖMS MODELIUOTI PROJEKTAS	27
3.1. FUNKCINIAI REIKALAVIMAI	27
3.2. VT MODELIAVIMO METODIKA	29
3.3. VARTOTOJO ŠAŠAJOS REIKALAVIMAI	31
3.4. NEFUNKCINIAI REIKALAVIMAI	32
3.5. SISTEMOS PROJEKTAS	33
3.5.1. <i>Sistemos elgsenos modelis</i>	34
3.5.2. <i>UML metamodelis ir jo praplėtimo aprašas</i>	35
3.5.3. <i>Roso veiklos taisyklių profilis</i>	38
4. SUKURTOJO ĮSKIEPIO EKSPERIMENTINIS ĮVERTINIMAS	41
4.1. EKSPERIMENTO KONCEPCIJA	41
4.2. DALYKINĖS SRITIES DUOMENŲ MODELIO SUDARYMAS	41
4.3. VEIKLOS TAISYKLIŲ SPECIFIKAVIMAS	42
4.4. PASIRENGIMAS EKSPERIMENTUI	43
4.5. EKSPERIMENTO SU SUKURTUOJU ĮSKIEPIU VT KŪRIMO EIGA	44
4.6. VT MODELIŲ TIKRINIMO FUNKCIONALUMO ĮVERTINIMAS	47
4.7. SUKURTOJO ĮSKIEPIO ĮVERTINIMAS	49
5. IŠVADOS	50
6. LITERATŪRA	51
7. MAGICDRAW UML PLUGIN DEVELOPMENT AND IMPLEMENTATION FOR BUSINESS RULES MODELING	53
8. PRIEDAI	54

Sutrumpinimų žodynas

OMG - (angl. Object management group) tarptautinės standartų organizacija.

UML - (angl. Unified modeling language) vieninga modeliavimo kalba.

OCL – (angl. Object Constraint language) objektų apribojimo kalba.

MDA - (angl. Model – driven architecture) modeliais pagrįsta architektūra.

CIM - (angl. computation independent model) nuo skaičiavimų nepriklausomas modelis.

PIM - (ang. Platform independent model) nuo platformos nepriklausomas modelis.

PSM - (angl. platform specific model) nuo platformos priklausomas modelis.

VT - veiklos taisyklė.

1. Įvadas

Šiuo metu pastebimas, inžinerinių kompiuterizuotų veiklos taisyklių principais grindžiamus metodus realizuojančių sprendimų, trūkumas. Vienas ryškiausių tokio pobūdžio metodų – tai Roso metodas, leidžiantis kurti formalius grafinius veiklos taisyklių (VT) modelius. Siekiant ištirti metodo pritaikomumą praktikoje, jis buvo modifikuotas, sukuriant veiklos taisyklių saugyklos metamodelį (Butleris; Kapočius 2002). Šiame darbe numatoma specifikuoti ir realizuoti veiklos taisyklių modeliavimo įrankį, padėsiantį panaudoti Roso metodą, unifikuotos modeliavimo kalbos UML pagrindu, veikiančiame kompiuterizuotos sistemų inžinerijos (CASE) įrankyje MagicDraw UML.

Darbe bus tyrinėjama veiklos taisyklių koncepcija ir jos taikymas projektuojant informacijos sistemas. Pagrindinis tyrimo objektas yra kompiuterizuotas grafinis veiklos taisyklių modeliavimas informacijos sistemų projektavimo stadijoje.

Tyrimo tikslas: atskleisti grafinio VT modeliavimo greta UML galimybes, sukuriant ir išbandant modifikuotu Roso metodu grindžiamą MagicDraw UML įrankio įskiepi (angl. *plugin*), kuris leistų modeliuoti veiklos taisykles, užmezgant ryšius tarp taisyklių ir esamomis MagicDraw UML priemonėmis sukurtojo duomenų modelio (klasių diagramos) elementų.

Tyrimo uždaviniai:

1. Išanalizuoti šiuos teorinius darbo pagrindus:
 - 1.1. Veiklos taisyklių koncepcijos agrondai, Roso metodas;
 - 1.2. MagicDraw UML praplėtimo galimybės;
2. Suformuluoti MagicDraw UML įskiepio kūrimo modelį;
3. Papildyti MagicDraw UML metamodelį, pritaikant jį Roso VT diagramoms saugoti;
4. Realizuoti MagicDraw UML įskiepi;
5. Taikant sudarytą VT modeliavimo metodiką, atlikti eksperimentinį bandymą realios dalykinės srities pagrindu;
6. Suformuluoti išvadas ir rekomendacijas tolimesniam metodikos vystymui.

Analizės tikslas

Analizės tikslas yra susipažinti su esamais problemos sprendimais, išanalizuoti Informacinių sistemų katedros (ISK) atliktus mokslinius darbus veiklos taisyklių taikymo srityje, išsiaiškinti, veiklos taisyklių formalizavimą pagal Roso metodą bei nustatyti sukurtojo Roso metodo metamodelio pritaikymo praplečiant UML metamodelį galimybes. Analizės metu taip pat siekta apžvelgti MagicDraw savybes ir išanalizuoti įskiepio kūrimo metodiką.

Analizės metodai

Pasirinktas mokslinės literatūros analizės metodas. Naudojantis šiuo metodu, buvo analizuojama mokslinė literatūra (publikacijos, moksliniai straipsniai, knygos, su sprendžiama problema susijusių metodų specifikacijos ir t.t.). Tiriant MagicDraw UML praplėtimo galimybes, remtasi autoriaus patirtimi, įgyta dirbant šio įrankio vystymo grupėje UAB „BPI“.

Darbo sudėtis

Be šio įvado darbą sudaro trys dėstymo skyriai, išvados, literatūros sąrašas ir 4 priedai. Antrajame darbo skyriuje pateikiama probleminės srities analizė, suformuluotos įskiepio kūrimo gairės. Trečiajame skyriuje pristatomas MagicDraw UML įskiepio projektas, sukurtasis UML metamodelio praplėtimas bei VT modeliavimo metodika. Ketvirtajame darbo skyriuje apžvelgtas sukurtasis prototipas, aptartas atliktas eksperimentas ir apibendrinti jo rezultatai.

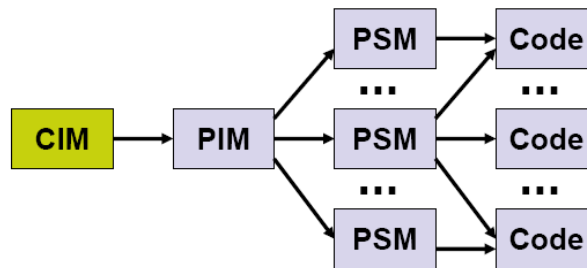
2. Kompiuterizuoto veiklos taisyklių modeliavimo IS projektavimo stadijoje analizė

2.1. Veiklos taisyklių vaidmuo IS kūrimo procese ir tyrimo koncepcija

Kuriant informacinių sistemų projektus yra susiduriama su daugeliu problemų. Pradedant per menku informacijos iš kliento gavimu ir baigiant projektuotojo ir kliento nesusišnekėjimu. Projektavimo etapo palengvinimui ir našumo bei kokybės didinimui yra kuriamos įvairios teorijos, metodai bei informacinių sistemų projektavimo įrankiai.

Viena iš lyderių informacinių sistemų specifikuojamose srityse OMG (angl. Object management group), pasiūlė specifikuojamą ir projektavimo architektūrą – MDA (OMG, 2007) (pav. 1.) (angl. Model – driven architecture), kuri projektavimo informacinės sistemos funkcionalumo specifikuojamą atskiria nuo realizavimo konkrečioje platformoje specifikuojamą.

Ši architektūra susideda iš 3 modelių: nuo skaičiavimų nepriklausomas modelis (veiklos modelis) - CIM (angl. computation independent model), toliau seka nuo platformos nepriklausomas modelis – PIM (ang. Platform independent model), toliau, nuo platformos priklausantis modelis – PSM (angl. platform specific model). Jei ši architektūra būtų realiai įgyvendinta ir galima būtų iš specifikuojamos (CIM) modelio automatizuotai gauti žemesnius projektavimo lygius ir jie būtų vienas nuo kito dalinai priklausomi, tai projektavimo našumas sparčiai padidėtų.



1 pav. MDA architektūra

Šiuo metu pasaulyje ir Lietuvoje vis aktyviau bandoma išplėsti CASE saugyklas, papildant jas veiklos taisyklių aspektu, kuris palengvintų sistemos projektavimą.

Šiame darbe bus bandoma palengvinti projektavimo etapą įvedant veiklos taisyklių modeliavimo funkcionalumą. T.y. bus bandoma CASE įrankį (MagicDraw UML), palaikantį UML specifikuojamą, papildyti veiklos taisyklių, formalizuotų pagal Roso metodą, modeliavimo priemone.

2.2. Aktualių sprendimų analizė

Kadangi žinių baze grindžiama informacinių sistemų inžinerija yra labai jauna IS inžinerijos šaka, tai nėra apibrėžtų tikslų standartų ir daugelis sprendimų yra specializuoti tik individualiam produktui (CASE įrankiui) ir jų naudojami algoritmai yra kūrėjų pramoninė paslaptis. Daugelis realizuotų VT taikymo įrankių yra orientuoti į konkrečią platformą (JAVA, C#, C kalbas) ar .Net technologijas. Visi sukurti sprendimai yra taikomi prie jau nusistovėjusių ir standartu tapusių UML, OCL ir t.t. modeliavimo kalbų ir jų modelių.

Veiklos taisyklių modeliavimas UML kalba

Yra daug būdų išreikšti veiklos taisykles UML modeliais, tačiau specialios notacijos nėra:

- Grafiškai naudojant objekto modeliuose asociacijos ir generalizacijos semantikas (Hagerty, 2005).
- Grafiškai naudojant apribojimus. Apribojimai leidžia tiksliau apibrėžti struktūrines taisykles (Pender, 2003).
- Grafiškai naudojant pastabą (simbolį), sujungtą su kitu elementu kaip asociacija ar atvejis.
- Rašydami teiginio logiką Objektų Apribojimo Kalba (OCL) (von Halle, 2001).
- Apibrėždami taisykles kaip kitų artefaktų ypatybės.

Problema yra tokia, kad UML neapibrėžia taisyklių kaip atskiro medelio kūrimo bloko - unikalios klasės modeliuoti elementą (Butleris; Kapočius, 2002).

Išskiriami tokie trūkumai kuriant veiklos taisykles UML diagramose (Butleris, Kapočius, 2001):

- Taisyklės yra išmėtytos UML modeliuose
- Sudėtinga nuspręsti, kaip ir kur taisyklė turi būti sumodeliuota
- Taisyklės pakartotinis naudojimas nėra lengvas
- Loginiai ryšiai tarp modeliavimo elementų yra neaiškūs
- Taisyklės neturi jokių ypatybių.

Nestandardizuoti VT modeliavimo sprendimai

Veiklos taisyklių užrašymo būdai taip pat yra skirtingi nuo rašymo formalia struktūrizuota kalba iki pateikiamų natūralios kalbos šablonų su ribotais veiksmų pasirinkimais (Leap Systems Company, 2006).

Veiklos taisyklės gali būti skirtingų tipų priklausomai nuo faktų skaičiaus kurie aprašo jų sudėtingumo lygį. Kai tam tikroms taisyklėms užtenka paprasčiausio „Jeigu-Tada“ (angl. if-then) principo, tai daugeliui kitų VT yra daug sudėtingesni išraiškos būdai. Todėl atsiranda būtinybė leisti VT aprašyti skirtingais tipais, kol garantuojama, kad matomumas - atvaizdavimas (angl. visability) nenukentės.

Labiausiai paplitusios taisyklės yra jeigu-tada (angl. IF-THEN), tai paprasčiausias veiklos taisyklės tipas, kuris turi sąlygą ir veiksmus. Kompanija „YASU Technologies“ yra realizavusi keletą netradicinių VT užrašymo būdų, tokių kaip VT sprendimų lentelė (angl. decision table) ir VT srautų diagramas (angl. Flow).

Sprendimų lentelės apjungia (angl. capture) susietas taisyklių lentelės „MS Excel“ lentelių pavidalu. Sprendimų lentelė YASU Technologies sukurtame „QuickRules.NET“ pakete yra unikalus sprendimas, kurį galima įvertinti „Decision table Engine“ technologija. Šis komponentas yra specialiai sukurtas įvertinti netgi labai didelėms sprendimų lentelėms per nykstamai mažą laiką (angl. no time) (Yasu Tech, 2007).

Taisyklės turinčios daug sąlygų (sąlygų seka) ir veiksmų yra geriausiai pateikiamos sprendimų lentelės forma. Ši forma yra priimtinausia siekiant pagerinti VT matomumui (angl. visibility) ir supratimui. Šioje lentelėje pateikiamas sąrašas sąlygų, kur įvertinami įvairūs parametrai, kurie užduodami VT.

VT srautų taisyklės priešingai nei taisyklės, sudarytos jeigu-tada (angl. if-then) principu ar sprendimų lentelės pavidalu, šis taisyklių formatas suteikia galimybę sudaryti VT kaip sprendimus (angl. decision) ir užduotis (angl. tasks) (Yasu Tech, 2007). T.y. taisyklės užrašomos kaip galimas sprendimų ir užduočių sąrašas. Šiuo atveju sprendimai aprašomi kaip sąlygų rinkinys, o užduotys aprašomos kaip veiksmai, kurie yra vykdomi priklausomai nuo tuo metu tenkinamų sąlygų. Toks sąlygų ir pasekmių (veiksmų) išskyrimas suteikia galimybę sudaryti kompleksines, sudėtines taisykles. Taip suteikiamas geresnis taisyklės supratimas bei tokia forma jos tampa geriau skaitomos. Priešingu atveju, jei šios kompleksinės taisyklės nebūtų užrašomos šiuo formatu, būtų sunku suprasti užduočių vykdymo seką. Šiuo atveju VT pateikiamos kaip nuoseklios vykdymo sekos. Panašiai kaip UML sekų diagramos.

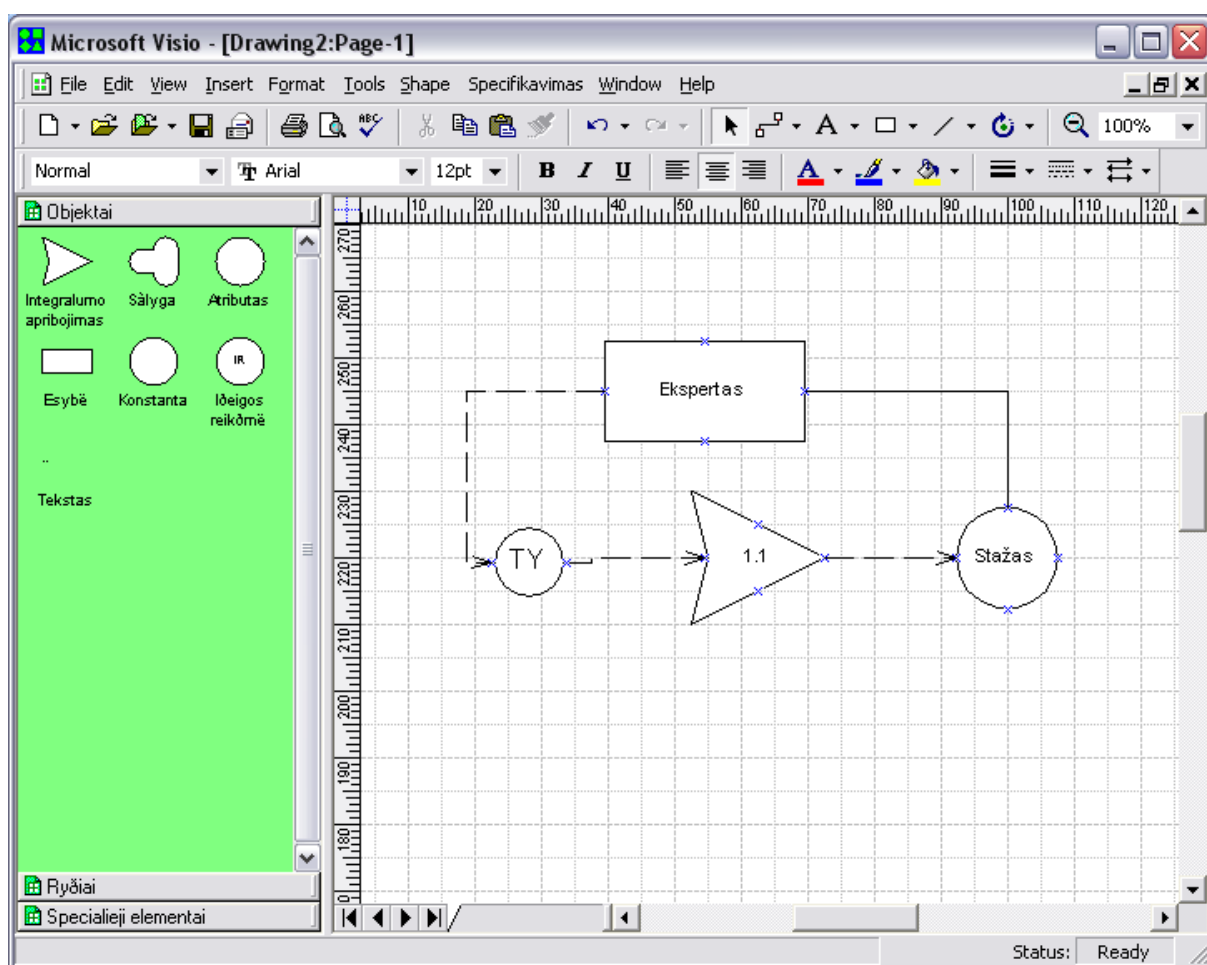
Šiuo principu sukurtų VT atvaizdavimas yra lengvai suprantamas. Jame tiesiogiai matosi, kas bus atliekama esant tam tikrom sąlygom. Pažvelgus į seką, veiklos analitikui matoma visos reikalingos VT ir lengvai suprantama VT vykdymo seka. „MS Visio“ paketas yra šio „QuickRules.NET“ metodo pagrindas.

Taigi veiklos taisyklės gali būti užrašomos įvairiais būdais ir įvairiomis priemonėmis. Kadangi yra sukurtų įvairių projektinių modelių kūrimo įrankių tai daugelyje įrankių yra stengiamasi suderinti su jau nusistovėjusiais standartais ir populiariais CASE įrankiais.

2.2.1. Grafinis veiklos taisyklių modeliavimas: Roso metodo realizacijos prototipas

Informacijos sistemų katedros magistranto sukurtas įrankis Visio pagrindu buvo vienintelis žinomas bandymas realizuoti Roso metode aprašytą VT grafinį modeliavimą.

Veiklos taisyklių įvedimas ir redagavimas jame realizuotas taip, kad būtų tenkinami visi formalūs baziniai reikalavimai (Kapočius, 2006). Įrankio darbo lango MS Visio aplinkoje pavyzdys pateiktas 2 pav.



2 pav. Esamo programinio įrankio VT diagramų braižymo aplinka

VT taisyklėms suteikti kodai, atitinkantys nurodytuosius aukščiau pateiktame sąraše. VT numeris rašomas greta VT tipo kodo. Svarbu pažymėti, kad jeigu vienai VT atvaizduoti reikalingos kelios atskiros taisyklės, jos numeruojamos skaičiumi iš dviejų dalių:

- skaičius iki taško – tai logiškai nedalomas VT numeris;
- skaičius po taško – tai sąlyginis šios VT ribose unikalus numeris.

Aiškumo dėlei greta diagramų vartotojui suteikiama galimybė įvesti ar redaguoti ir taisyklės išraišką natūraliąja kalba. Realioje specifikacijoje ši išraiška padeda vartotojui greičiau perprasti diagramos esmę. Dirbant Visio aplinkoje, VT tipai ir duomenų modelio elementai yra pasirenkami iš saugomų saugyklos duomenų bazės. Roso metode numatytų specialiųjų elementų rinkinys yra nekintamas, tačiau kiekvienas elementas privalo turėti ir užregistruotą atitikmenį saugyklos duomenų bazėje.

Įrankis nėra pilnai išbaigtas. Veiklos taisyklės nėra susietos su UML ar kitų plačiai naudojamų duomenų struktūros modelių elementais. Kiti paminėtini trūkumai:

- nėra diagramų sintaksės tikrinimo (teisingumas pakankamai neužtikrinamas nei braižymo metu, nei išsaugant diagramą);
- įrankis orientuotas į Roso notacijos išbandymą, todėl jis negarantuoja teisingo nubraižytų diagramų išsaugojimo taisyklių saugykloje, neužtikrintas ir teisingas jau išsaugotų taisyklių atkūrimas.

2.3.Roso metodas

2.3.1. Metodo koncepcija ir sudėtis

Metodas yra pasiūlytas Ronaldo Roso (*Ronald Ross*) (Ross, 1997). Galima teigti, jog Roso VT modelis buvo kuriamas kaip detali nestruktūrinio pobūdžio veiklos taisyklių studija.

Pagal Roso metodą taisyklės gali būti *nedalomos* ir *išvestinės*. Yra išskiriami 32 nedalomų taisyklių tipai, kurie grupuojami į 7 tipų grupes. Išvestinė taisyklė – tai taisyklė, kuri išreiškiama kitų taisyklių aprašais. Išvestinės taisyklės nėra nedalomos ir gali būti sudarytos iš keleto nedalomų taisyklių arba kitų išvestinių taisyklių. Yra išskiriami 58 išvestinių taisyklių tipai, kurie pagal panaudojimo sritį yra grupuojami į 12 tipų grupių. Būtina pažymėti, kad ypač daug dėmesio R. Rosas skiria taisyklių tipų aprašymui, todėl šis modelis klasifikavimo požiūriu yra ypač tikslus.

Nedalomos taisyklės yra skirstomos į septynias tipų grupes (Ross, 1997). Taisyklės gali būti sujungiamos įvairiomis kombinacijomis tokiu būdu gaunant *sudėtines* taisykles.

Išvestinės taisyklės – tai taisyklės, kuri išreiškiama kitų taisyklių aprašais. Išvestinės taisyklės nėra nedalomos. Išvestinė taisyklė gali būti sudaryta iš keleto nedalomų taisyklių arba kitų išvestinių taisyklių. Šio tipo taisyklėms būdingos nedalomų taisyklių savybės, iš jų ir specifinio tipo išėties vertė.

Be detalios taisyklių klasifikavimo sistemos, R.Roso metode yra pasiūlyta ir unikali VT modeliavimo užrašant jas grafiniu pavidalu metodika. Formaliai užrašant VT reikia remtis

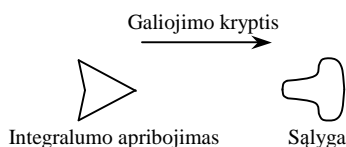
dalykinės srities duomenų modelių. Koks būtent modelis turėtų būti naudojamas, R. Rosas nenurodo, tačiau yra apibrėžta, kokie duomenų tipai turi būti žinomi. Pagrindiniai jų yra esybės, atributai, asociacijos, sąryšiai, potipiai. Taisyklės teiginio užrašymas formaliu pavidalu susideda iš 6 pagrindinių etapų:

1. *Nustatymas, ar VT yra integralumo apribojimas, ar sąlyga.* Kitaip tariant, reikia nustatyti, ar taisyklė reikalauja vykdymo, ar yra išreikšta tik kaip patikrinimas.

Integralumo apribojimas – tai taisyklė, kurios rezultatas visada privalo būti loginis vienetas (loginė tiesa). Kadangi tokios taisyklės neturi duoti neigiamo rezultato, todėl privalo būti vykdomos.

Sąlyga – tai taisyklė, kurios rezultatas gali būti arba loginis vienetas (loginė tiesa), arba loginis nulis (loginis melas). Rezultatas taip pat gali būti ir nežinomas. Šios taisyklės naudingos, kuomet reikia nustatyti, kokias taisykles vykdyti arba kokius testavimus atlikti (t.y., kokias kitas taisykles iškviešti) atsižvelgiant į tam tikrą situaciją. Tokias taisykles galima išdėstyti nuosekliai, nurodant, kas turi vykti, jeigu aukštesnio lygio taisyklė yra tenkinama.

Modeliuojant taisykles apribojimams ir sąlygoms atskirti naudojami specialūs žymėjimai (3 pav.). Simboliai gali būti sukami į bet kurią pusę, tačiau būtina nepamiršti jų galiojimo krypties.



3 pav. Taisyklės vykdymo ir taisyklės tikrinimo žymėjimas (Butleris; Kapočius, 2002)

2. *VT bazės nustatymas.* Kitaip tariant, reikia nustatyti, kuriam objektui, vadinamam *baze*, reikėtų “priskirti” nagrinėjamą VT. Dažniausiai bazė – tai duomenų modelio objektas (esybė, atributas ir pan.), tačiau tai gali būti ir kita VT ar jos išėigos reikšmė. Taisyklė gali būti traktuojama kaip jos bazės savybė.

Bazę privalo turėti kiekviena VT, nes tai yra svarbiausias elementas interpretuojant taisyklę. Bazės egzemplioriai gali būti įtraukti ir į taisyklės atliekamą patikrinimą (Ross, 2003). Toks patikrinimas atliekamas su kiekvienu bazės egzemplioriumi.

3. *Korespondento(-ų) nustatymas.* Visos VT be bazės paprastai apima dar bent vieną duomenų modelio objekto egzempliorių. Tokie objektai vadinami *korespondentais* ir jie yra būtini kiekvienoje taisyklėje. Korespondentai visuomet yra svarbūs atliekant taisyklės tipo apibrėžtą testą. Korespondentų gali būti ne tik duomenų modelio objektas, bet ir kita taisyklė, jos išėigos reikšmė arba veiksmas.

Kiekviena taisyklė turi vieną ar daugiau korespondentų. Galima išskirti šias ypatybes:

- Nedalomos taisyklės, priklausančios *tipo patvirtinimo* tipų grupei, privalo turėti du arba daugiau korespondentų, nes veikia kaip loginiai operatoriai IR ar ARBA.
- Nedalomos taisyklės, priklausančios *matematinio įvertinimo* tipų grupei, gali turėti daugiau negu vieną korespondentą.
- Likusioms tipų grupėms priklausančios nedalomos taisyklės gali turėti tik vieną ir ne daugiau korespondentų.

Tiek išeinantis iš bazės ryšys, tiek ir einantys į korespondentus ryšiai VT diagramoje visada vaizduojami punktyrine rodykle.

4. Taisyklių tipų parinkimas. Parinkti taisyklės tipą (arba nustatyti, kokius testus ji atlieka) yra būtina, nes tik esant nurodytam tipui galima teisingai interpretuoti taisyklę. Taisyklė gali priklausyti tiek nedalomų VT, tiek išvestinių VT grupės tipui. Visi taisyklių tipai tinka tiek integralumo apribojimams, tiek ir sąlygoms. VT diagramoje tipo santrumpa įrašoma apribojimo arba sąlygos simbolio viduje.

5. Bazės ir korespondento(-ų) susiejimas. Dažnai svarbu išskirti ir teisingą sąryšį tarp bazės ir korespondento. Šis sąryšis privalo sekti iš duomenų modelio. Galima priimti, jog duomenų modelis yra sudarytas iš *terminų* ir *faktų* apie probleminę sritį. Faktai gali būti šių kategorijų:

- Sąryšio tipai, pvz.: “klientas pateikia užsakymą”
- Potipių sąryšiai, pvz.: “automobilis yra prekė”
- Atributų sąryšiai, pvz.: “automobilis turi registracijos numerį”

Svarbu pabrėžti, jog VT niekuomet negali įvesti naujo fakto (sąryšio). Visi svarbūs faktai (sąryšiai) turi būti atspindėti duomenų modelyje.

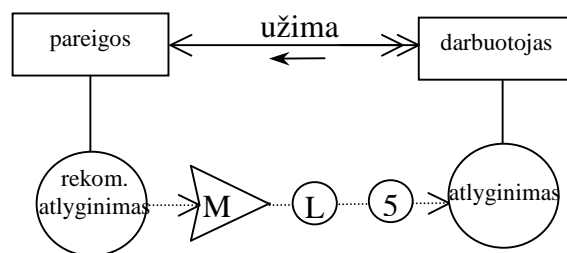
Taigi, kiekvienas sąryšis tarp taisyklės bazės ir korespondento(-ų) turi būti paimtas iš duomenų modelio. Būtina išskirti ir kitą prielaidą: kiekvienas pavaizduotas sąryšis (fakto tipas) tarp taisyklės bazės ir korespondento(-ų) yra laikomas svarbiu taisyklės interpretacijai. Taigi, nesvarbūs sąryšiai į diagramą neturi būti įtraukiami (Butleris; Kapočius, 2002).

6. Papildomų VT modelio ypatybių nustatymas. Roso notacijoje yra numatyta galimybė naudoti papildomus simbolius, galinčius pakeisti standartinę tam tikro tipo taisyklės interpretaciją. Šie simboliai skirstomi į dvi pagrindines grupes:

1. Interpretatoriai – tai simboliai, nurodantys, jog taisyklę jos bazės atžvilgiu reikia interpretuoti ne taip, kaip reikėtų remiantis tik jos tipu. Šie simboliai yra žymimi ant bazės ryšio rodyklės.
2. Apribojimai – tai simboliai, nurodantys, jog taisyklei jos korespondento(-ų) egzempliorių atžvilgiu turi būti taikomi specialūs apribojimai.

Be specialiųjų simbolių dar gali būti vartojamos konstantos, taisyklių išeišgos reikšmės ir kitos papildomos priemonės (Butleris; Kapočius, 2002). Paprastai taisyklės išeišgos reikšmė yra “nematoma” ir vartojama taisyklės viduje tam tikram testavimui atlikti arba apribojimui pritaikyti. Tačiau tam tikrose taisyklėse reikia tiesiogiai tikrinti kitų taisyklių išeišgos reikšmės (tai labiau būdinga sąlygos tipo taisyklėms). Tuomet VT išeišgos reikšmė atsispindės ir taisyklės, vartojančios šią reikšmę, diagramoje. Kaip jau buvo minėta, taisyklės išeišgos reikšmė gali būti kitos taisyklės baze arba korespondentu.

Paprastos taisyklės pavyzdys pateiktas 4 pav. Ji apibrėžia tokį apribojimą: *“Rekomenduojamas pareigų atlyginimas turi būti mažesnis už bent penkių darbuotojų, užimančių šias pareigas, atlyginimą”*. Šios taisyklės bazė yra duomenų modelio esybės pareigos atributas rekom.atlyginimas (rekomenduojamas atlyginimas). Taisyklės korespondentas – duomenų modelio esybės darbuotojas atributas atlyginimas. Mus domina tik tos pareigos, kurias užima darbuotojai, todėl į diagramą įtrauktas ir šis duomenų modelio sąryšis (faktas). Taisyklės tipas yra **Mažiau-už** (kodas MU). Diagramoje taip pat naudojami ir du apribojimai: žemesniojo slenksčio apribojimas (simbolizuoja rutuliukas su raide L) ir numeratorius (simbolizuoja rutuliukas su skaičiumi 5).

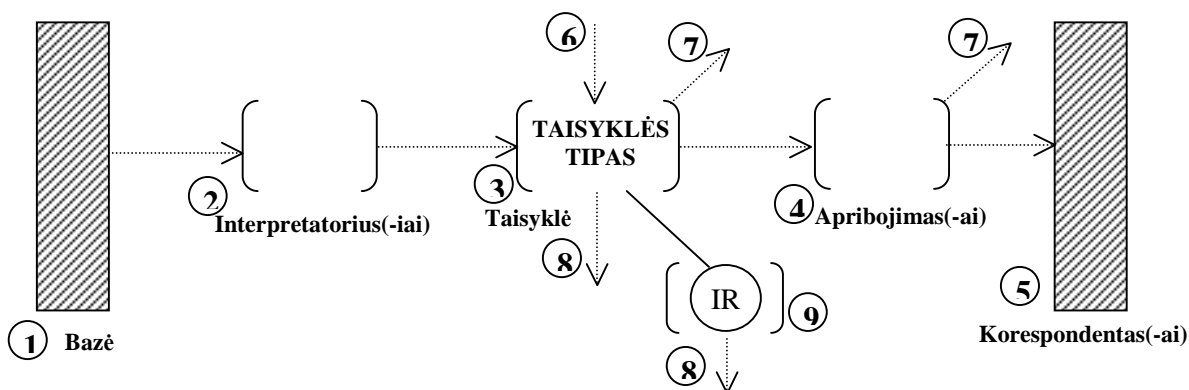


4 pav. Veiklos taisyklės grafinio modelio iliustracija (Butleris; Kapočius, 2002)

Taigi, galima teigti, jog Roso metodas leidžia atlikti išsamų VT struktūrizavimą pagal tipus. Roso VT modeliavimo metodika užtikrina vienareikšmišką VT susiejimą su duomenų modelio elementais, apribojamais VT ar tiesiog svarbiais jos interpretacijai. Be to, Roso VT modeliai leidžia vizualizuoti VT ir duomenų modelio sąsają (Butleris; Kapočius, 2001). Griežta notacija sukuria prielaidas saugoti diagramas duomenų bazėje, tačiau Roso neapibrėžia loginio modelio, kuriuo remiantis būtų galima tokią bazę sukurti. Tai traktuotina kaip metodo trūkumas.

2.3.2. Veiklos taisyklių formalizavimas pagal Roso metodą

VT formalizavimą reikėtų pradėti tuojau pat po taisyklės teiginio preliminarus patvirtinimo. Šiame etape naudojama Roso modelyje apibrėžta taisyklių užrašymo specialių diagramų pavidalu metodika. Priimama, jog kiekviena pagal Roso metodą sumodeliuota VT pasižymi tokia pačia struktūra, kuri pateikta 4 pav.



5 pav. Veiklos taisyklės formalizavimo pagal Roso metodą sintaksė (Butleris; Kapočius, 2002)

Veiklos taisyklės diagramos sudedamosios dalys yra šios (žr. 5 pav.):

1. *Privaloma*. Gali būti: duomenų modelio tipas, į duomenų modelyje neapibrėžta konstanta, kita VT arba kitos VT išeigos reikšmė.
2. *Nebūtina*. Teisingas panaudojimas priklauso nuo VT tipo.
3. *Privaloma*. Gali būti integralumo apribojimas arba sąlyga.
4. *Nebūtina*. Teisingas panaudojimas priklauso nuo VT tipo.
5. *Privaloma*. Gali būti: duomenų modelio tipas, į duomenų modelyje neapibrėžta konstanta, kita VT arba kitos VT išeigos reikšmė.
6. *Nebūtina*. Vaizduoja įeigos taisyklę.
7. *Leistina daugiau negu vienas*. Taisyklė gali turėti daugiau negu vieną korespondentą.
8. *Nebūtina*. Vaizduoja išeigos taisyklę.
9. *Nebūtina*. Vaizduoja taisyklės išeigos reikšmę.

Be šių elementų VT diagramoje turi matytis ir visi jos interpretacijai svarbūs duomenų tipai (Butleris; Kapočius, 2002).

Formalizuojant VT būtina laikytis šių principų:

1. *Į taisyklės diagramą turi būti įtraukti tik tie duomenų tipai, kurie yra vienareikšmiškai svarbūs specifikuojamai taisyklei.*

- Naudojant tik vienareikšmiškai svarbius duomenų tipus, specifikuojama taisyklė bus tiksliau interpretuojama bei bus išvengta nesusipratimų, kuriuos galėtų sukelti pertekliniai VT diagramoje matomi elementai.

- VT diagramoje turi būti pateikti visi duomenų tipai, kurie susieti su formalizuojamu taisyklės teiginiu reikalavimų specifikavimo pagal siūlomą metodą metu. Didelė tikimybė, jog šis tipų rinkinys bus išsamus, tačiau projektuotojas privalo įsitikinti, ar nėra trūkstamų komponentų.
 - VT modeliavimo metu naudojamas programinis įrankis turėtų užtikrinti VT diagramoje naudojamų duomenų tipų atitikimą bazėje saugomai struktūrai bei neliesti įvesti ligiška neišsamų diagramų (pvz.: naudojančių atributų simbolių, neturinčius sąsajų su diagramoje matomomis esybėmis, arba ryšius tarp esybių neįtraukiant pačių esybių ir pan.).
2. *Būtina laikytis principo, jog interpretuojant taisyklę ji bus taikoma tik tiems korespondento egzemplioriams, kurie susiję su bazės egzemplioriumi.*
- Kiekvienas į taisyklės diagramą įtrauktas ryšys tarp bazės ir korespondento(-ų) turi būti svarbus taisyklės interpretacijai. Nesvarbūs ryšių tipai į diagramą neįtraukiami.
3. *Taisyklės turi būti tiksliai identifikuotos.*
- Veiklos taisyklės ir jas atitinkančios diagramos turi būti identifikuojamos pagal tam tikrą sistemą parinktais vienodais identifikatoriais (pvz.: visos taisyklės gali būti numeruojamos, tą patį numerį priskiriant tiek taisyklės teiginiui, tiek ir jos diagramai). Rekomenduojama naudoti reikalavimų specifikavimo metu VT suteiktus kodus.
4. *Kiekviena VT gali būti priskirta tik vienam tipui.*
5. *Kiekviena VT diagramoje vaizduojama nedaloma taisyklė gali turėti tik vieną bazę.*
6. *Taisyklė turi būti specifikuojama kaip įmanoma bendresniam duomenų modelio elementui.*
- Specifikuojant taisyklę reikia išskirti tokį bazės tipą, kad jis apimtų visus taisyklės apibrėžiamus potipius. Priimama, jog visi šio tipo potipiai paveldi šią taisyklę. Taigi, kiekvienam iš potipių išskirti atskiros taisyklės nereikėtų.
 - Būtina įsitikinti, ar tikrai visi parinktos bazės potipiai paveldi taisyklę. Taip pat potipiui negali būti išskirta taisyklė, kuri pažeidžia bendresniam tipui, į kurį įeina nagrinėjamas potipis, taikomą taisyklę.
7. *Taisyklės Įgalintojai (ar kitų tipų taisyklės) neturėtų būti painiojamos su trigeriais, naudojamais duomenų bazių valdymo sistemoje.*
- Trigeris gali būti naudojamas iškviesti tam tikrai procedūrai, pritaikančiai tam tikrą taisyklę, kuomet įvyko tam tikras įvykis. Tai tik viena nedidelė taisyklės realizavimo proceso dalis. VT neregamentuojama, kaip ar kada programiniu požiūriu jos turėtų būti vykdomos. VT reglamentuoja pačią veiklą ir tai, kas svarbu šios veiklos dalyviams.

2.3.3. Formalūs reikalavimai veiklos taisyklėms

Remiantis Roso metodu ir pasiūlymo specifika suformuluoti baziniai reikalavimai formaliai užrašytoms VT pateikti žemiau (Butleris; Kapočius, 2002).

1. Veiklos taisyklė privalo būti priskirta vienam ir tik vienam tipui.

$$\forall x (\text{Taisyklė}(x) \rightarrow \exists ! k (\text{Tipas}(k) \cap \text{charakterizuoja}(k, x)))$$

2. Veiklos taisyklė privalo turėti vieną ir tik vieną bazę.

$$\forall x (\text{Taisyklė}(x) \rightarrow \exists ! y (\text{Bazė}(y) \cap \text{yra_pagrindas}(y, x)))$$

3. Veiklos taisyklės bazė gali būti arba duomenų tipas, arba konstanta, arba kita VT, arba kitos VT išeigos reikšmė.

$$\begin{aligned} & \forall x \forall y (\text{Taisykle}(x) \cap \text{Bazė}(y) \cap \text{yra_pagrindas}(y, x)) \rightarrow \\ & \rightarrow (\text{Duomenų_tipas}(y) \cup \text{Konstanta}(y) \cup (\text{Taisyklė}(y) \cap (x \neq y)) \cup \\ & \cup (\text{Išeigos_reikšmė}(y, x') \cap \text{Taisyklė}(x') \cap (x' \neq x))) \end{aligned}$$

4. Veiklos taisyklė privalo turėti ne mažiau negu 1 korespondentą.

$$\forall x (\text{Taisyklė}(x)) \rightarrow \exists z (\text{Korespondentas}(z) \cap \text{tikrina}(z, x))$$

5. Veiklos taisyklės korespondentas gali būti arba duomenų tipas, arba konstanta, arba kita VT, arba kitos VT išeigos reikšmė.

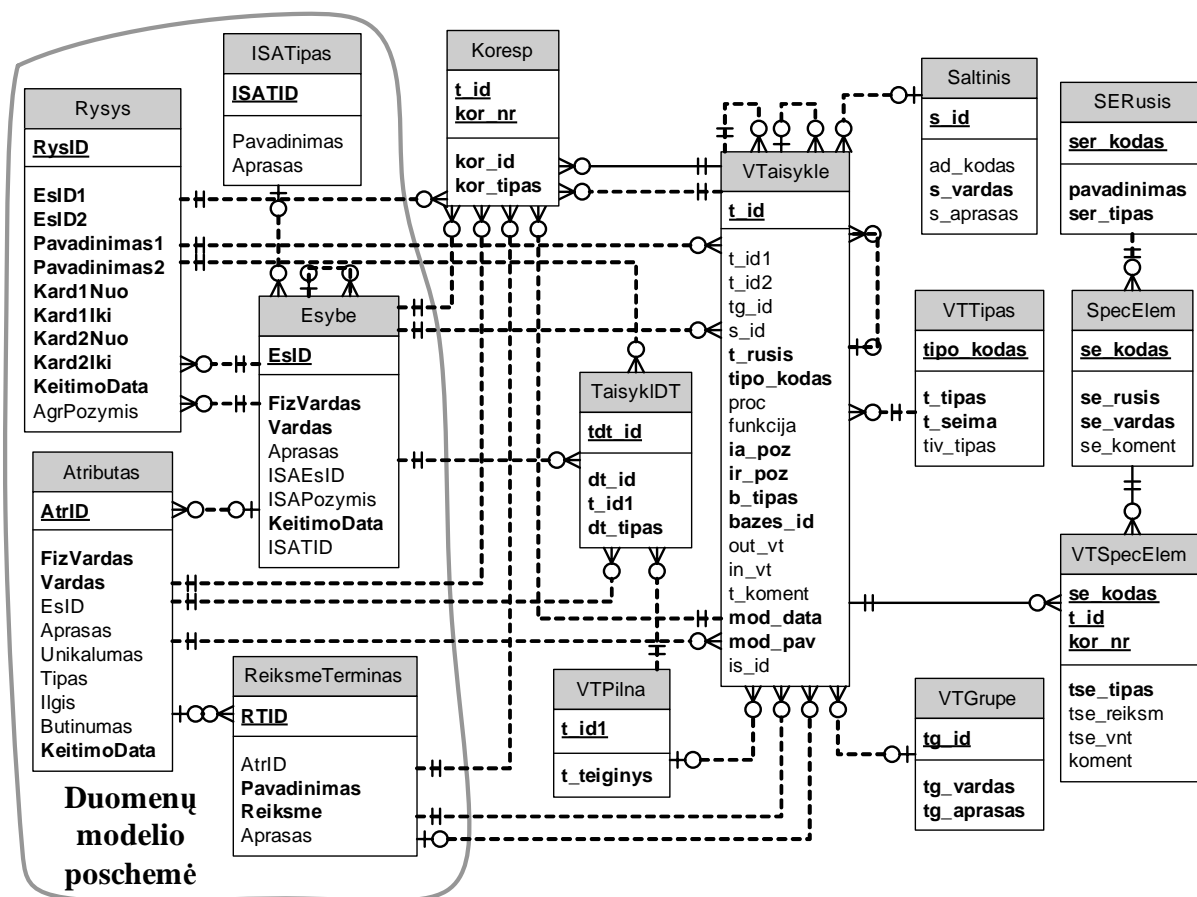
$$\begin{aligned} & \forall x \forall z (\text{Taisykle}(x) \cap \text{Korespondentas}(z) \cap \text{tikrina}(z, x)) \rightarrow \\ & \rightarrow (\text{Duomenų_tipas}(z) \cup \text{Konstanta}(z) \cup (\text{Taisyklė}(z) \cap (x \neq z)) \cup \\ & \cup (\text{Išeigos_reikšmė}(z, x') \cap \text{Taisyklė}(x') \cap (x' \neq x))) \end{aligned}$$

2.3.4. Pagal Roso metodą formalizuotų veiklos taisyklių saugyklos modelis

Saugykloje saugoma tik VT diagramoje atsispindinti informacija. Saugyklos loginis modelis pateiktas 6 pav.

Kiekviena logiškai vientisa VT, išreiškiama viena diagrama, VTPilna gali susidėti iš keleto veiklos taisyklių VTaisykle, priklausančių vienam iš Roso metode numatytų tipų VTTipas. Kiekviena pilna taisyklė gali būti susijusi su daugeliu dalykinę sritį charakterizuojančių duomenų modelio elementų iš lentelių Esybe, Rsys, Atributas, ReiksmeTerminas. Kiekviena veiklos taisyklė VTaisykle, turi bazę ir vieną ar daugiau korespondentų Korespondentas. Korespondentu, kaip ir bazę, gali būti duomenų modelio elementas, konstanta kita VT VTaisykle arba kitos VT išeišgos reikšmė. Veiklos taisyklės VTaisykle diagramoje gali būti naudojami vienas ar daugiau specialiųjų simbolių SpecElem, kurie sugrupuoti pagal rūšį SERusis.

Be tiesiogiai VT diagramoje atsispindinčios informacijos yra saugomi ir papildomi duomenys apie taisyklės kilmės šaltinį Saltinis, be to, taisyklės gali būti grupuojamos į įvairaus pobūdžio grupes VTGrupe.



6 pav. Modifikuotu Roso metodu formalizuotų VT saugyklos loginės struktūros modelis (Butleris; KAPOČIUS, 2002)

2.3.5. Veiklos taisyklių struktūrizavimo eiga

VT struktūrizavimo veiklų sekų diagrama pateikta 7 paveiksle. Galima išskirti šiuos etapus:

1. Sisteminės informacijos įvedimas. Registruojami duomenys, kurie žinomi iš anksto ir nesikeis. Tai duomenys, aprašyti Roso metode. Etapo metu pildomos šios VT saugyklos loginio modelio lentelės:

1.1. lentelės *SERusis*, *SpecElem*;

1.2. lentelė *VTTipas*.

Turi būti įvesti visi specialūs elementai ir taisyklių tipai, net jeigu dalis jų ir nėra naudojama VT diagramose.

2. Papildomos informacijos įvedimas. Ši informacija gali būti įvedama ir vėliau, bet pageidautina ją turėti prieš įvedinėjant VT. Registruojant papildomą informaciją pildoma:

2.1. lentelė *Saltinis*;

2.2. lentelė *VTGrupe*;

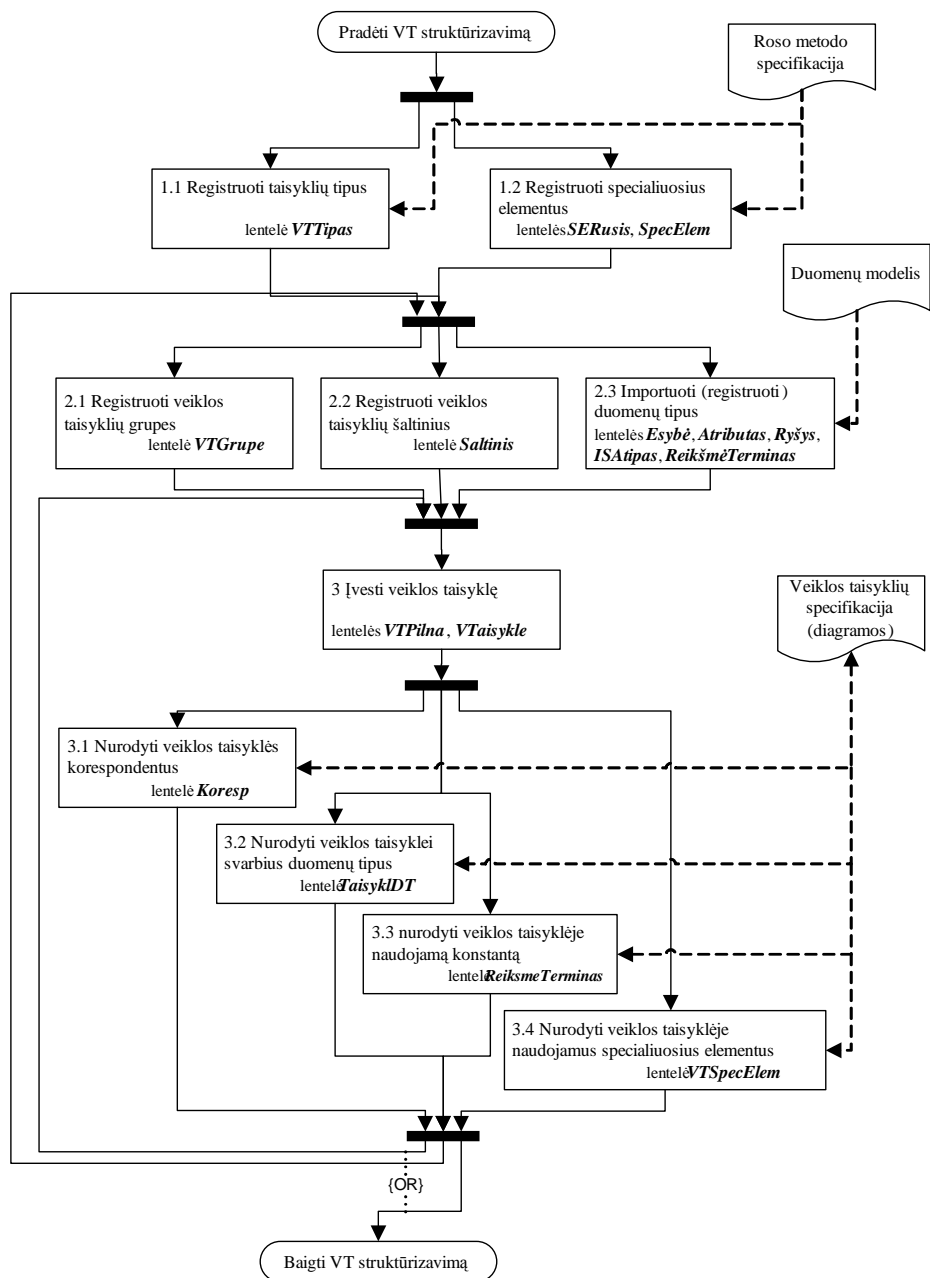
2.3. lentelės *Esybe*, *Atributas*, *Rsys*, *ISATipas*, *ReiksmeTerminas*.

Projektavimo stadijoje, kurią ruošiamės kompiuterizuoti, jau yra žinomas koncepcinis duomenų modelis, pagal kurį sudaromas loginis duomenų modelis. Šio modelio elementai turėtų būti sukelti į VT saugyklą arba importuojami į ją. Taikant reikalavimų specifikavimo metodą duomenų modelis gali būti be praradimų automatizuotai importuojamas į VT saugyklą. Vis tik net ir projektavimo stadijoje tam tikri duomenų modelio elementai gali keistis ir saugyklos duomenų modelio poschemės lentelės turėtų būti papildomos. Ypač svarbu užtikrinti, kad prieš įvedant VT, visi jai svarbūs duomenų modelio elementai jau būtų įvesti.

3. Veiklos taisyklės įvedimas. Sukuriama ir įvedama VT diagrama. Pildomos lentelės *VTPilna*, *VTaisykle*, *Koresp*, *ReiksmeTerminas* (įvedant naują VT naudojamą konstantą) ir pagalbinės lentelės *TaisyklDT*, *VTSpecElem*.

Papildomos informacijos ir VT įvedimas gali būti atliekamas lygiagrečiai.

Atsižvelgiant į duomenų įvedimo eiliškumą ir loginius sąryšius tarp lentelių turėtų būti projektuojama ir grafinė vartotojo sąsaja (Butleris; Kapočius, 2002).

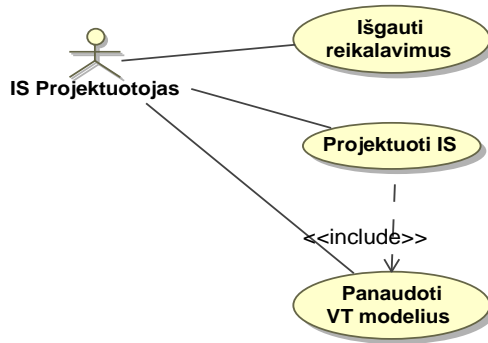


7 pav. Veiklos taisyklių saugyklos pildymo veiklos diagrama (Butleris, Kapočius, 2002)

2.4. Veiklos taisyklių modeliavimo įrankio vartotojų analizė

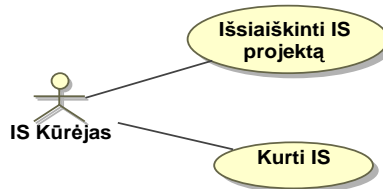
Vartotojų aibė, tipai ir savybės

Kuriamo produkto vartotojų aibė susideda iš: IS projektuotojų, IS realizuotojų ir IS užsakovų. IS projektuotojai (pav. 8) - specialistai, kurie gerai žino įvairias projektavimo kalbas ir jomis aprašo specifikacijoje sukauptas probleminės srities žinias bei laukiamus rezultatus iš kuriamos IS.



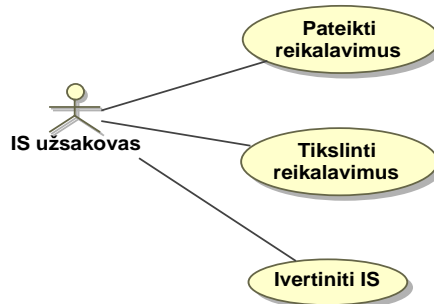
8 pav. IS Projektotojo veiksmai IS projektavime

IS realizuotojai (pav. 9) – tai specialistai, kurie gerai žino įvairias projektavimo kalbas ir skaitydami IS specifikaciją pateiktą šia kalba, ją realizuoja (parašo programos kodą, sujungia aparatūra ir t.t.).



9 pav. IS realizuotojai veiksmai IS kūrime

IS užsakovas (pav. 10) – tai fizinis ar juridinis asmuo, kuris dažniausiai neturi jokių IS projektavimo kalbų žinių arba jos būna labai menkos ir ribotos, tačiau jis bando perteikti norimą kuriamos IS funkcionalumą IS projektuotojui.



10 pav. IS užsakovo veiksmai IS projektavime

Vartotojų tikslai ir problemos

IS projektuotojų tikslas yra surinkti kuo daugiau probleminės srities žinių iš užsakovo, jas tiksliai suspecifikuoti natūralia kalba klientui ir projektavimo kalba IS kūrėjui (Morgan, 2002).

IS užsakovo tikslas yra konkrečiai išdėstyti norimą IS funkcionalumą ir kruopščiai išnagrinėti projektuotojo pateiktą specifikaciją natūralia kalba prieš ją pateikiant realizacijai.

IS kūrėjų tikslas yra sukurti informacinę sistemą, kuri atitiktų specifikaciją. Informacinė sistema dažniausiai yra kuriama iš specifikacijos pateiktos formalizuota projektavimo kalba (UML). Galimas variantas, jog projektuotojas modeliuos tik tas veiklos taisykles, kurios yra identifikuotos kaip kritinės ir tiek kūrimo metu, tiek ir eksploatuojant būsimąją sistemą, turės tendenciją dažnai keistis arba tikėtini jų pasikeitimai bus labai svarbūs užtikrinant nuoseklią sistemos veiklą.

Dažniausiai kyla problemos dėl kliento ir projektuotojo nesusišnekėjimo ir nevienodo konkrečių teiginių interpretavimo, dėl ko yra gaunama informacinė sistema, kuri neatitinka užsakovo reikalavimo ir ją reikia taisyti (Pilone; Pitman 2005). Kadangi reikia taisyti programą, kuri buvo blogai specifikuota, tai kuo vėliau yra pastebimi netikslumai tuo IS kūrėjai turi daugiau darbo, kad problema būtų išspręsta, kas labai padidina IS kūrimo kaštus.

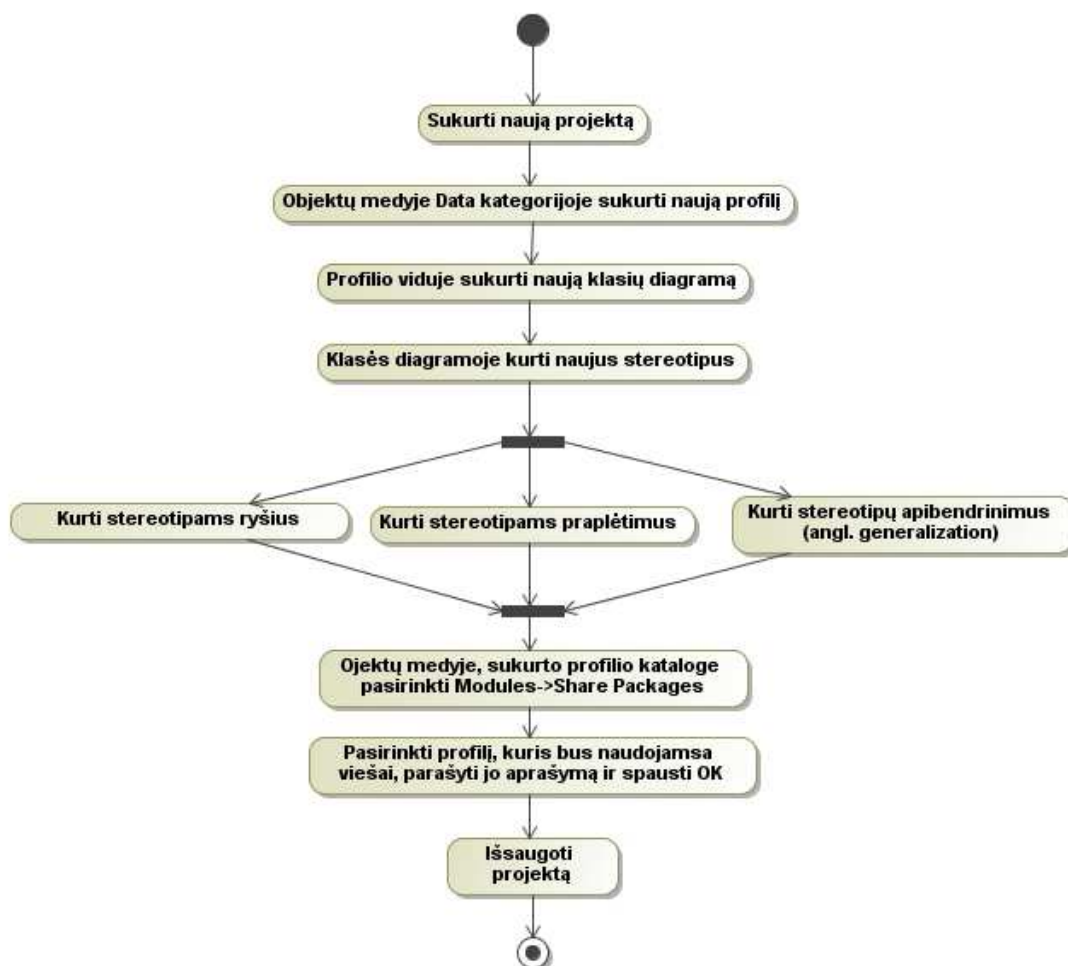
Taigi šis darbas yra skirtas palengvinti ir pagreitinti informacinių sistemų projektavimą. Taigi iš kuriamo įrankio daugiausiai naudos turės IS projektuotojai ir IS kūrėjai.

2.5.MagicDraw UML produkto praplėtimo galimybių analizė

Kuriamos sistemos projektavimui bus naudojamas MagicDraw UML paketas, kuris jau eilę metų yra pripažįstamas vienas geriausių projektavimo įrankių.

2.5.1. MagicDraw UML profilio kūrimas

Profilis gali būti sukurtas bet kuriame darbiname projekte, tačiau vienas profilis gali būti naudojamas daugelyje projektų. Profilio pakartotinam panaudojimui yra būtina jį sukurti su viešais duomenimis vadinamais moduliais. Profilio kūrimo eiga pateikta 11 pav.



11 pav. MagicDraw UML profilių kūrimo veiklos diagrama

Kuriamame profilyje dažniausiai yra kuriami konkrečių modelio elementų praplėtimai – stereotipai (No Magic, 2007).

Kuriami nauji, stereotipizuoti modelio elementai gali turėti:

- papildomus atributus (angl. tag),
- gali būti paslėpti esantys standartiniai atributai,
- specifinę elemento ikoną,
- specifiniu elementų ribojimus (elementų jungimo taisykles).

Sukurti profiliai yra saugojami ir gali būti užkraunami, tik kai reikalingas specifinis modelis, kuris yra kuriamas tik vieną kartą.

MagicDraw UML specifinių diagramų kūrimas

Kiekvienam naujam profiliui dažniausiai yra kuriama specifinė diagrama. MagicDraw UML pakete yra realizuotas diagramų (modelių) kūrimo įrankis, kurio pagalba yra sukuriamos atitinkamos įrankių juostos, kuriose yra įdedami specifiniai, profilyje aprašyti objektai, modelio elementai.

Naudojantis sukurtomis įrankių juostomis realizuojamas vizualus specifinio profilio pateikimas, kuris pagerina modelio supratimą ir naudojamumą. Patogus ir aiškus modelio

skaitymas ir sudarymas lemia vartotojų palankumą sukurtam modeliui ir daro jų darbą efektyvesnį.

2.5.2. MagicDraw įskiepis

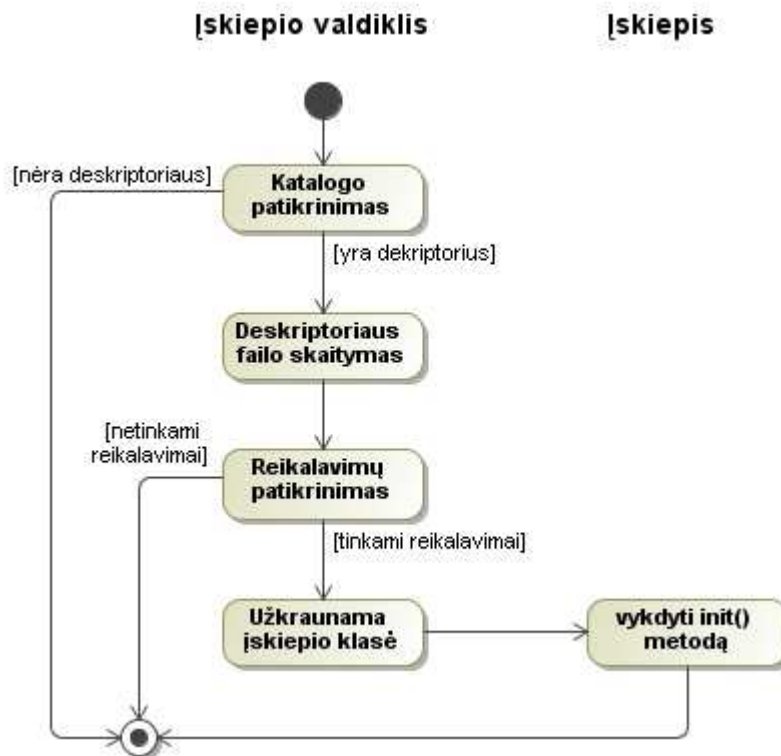
Mūsų universitetas bendradarbiauja su įmone No Magic, kuri kuria projektavimo įrankį MagicDraw UML, kuris jau kelis metus iš eilės yra pripažįstamas geriausiu savo srityje.

Kuriamas įskiepis bus realizuotas MagicDraw UML programiniam paketui. Gamintojai šiam paketui paliko galimybę susikurti paketo įskiepi, kuri galima integruoti su visu programinio paketo funkcionalumu. Aprašymas, kaip sukurti plėtinį yra kiekviename kompiuteryje, kuriame yra įdiegtas MagicDraw UML paketas (No Magic, 2007).

MagicDraw įskiepio veikimas

MagicDraw kiekvienu pasileidimo metų „plug-ins“ kataloge ieško pakatalogių (pav. 12):

- Jei pakatalogyje yra įskiepio deskriptoriaus failas, tai įskiepio tvarkytojas nuskaito šį failą.
- Jei deskriptoriaus failo aprašyti reikalavimai yra tenkinami, tai įskiepio tvarkytojas užkrauna apibrėžtą klasę (apibrėžta įskiepio klasė turi būti `com.nomagic.magicdraw.plugins.Plugin` poklasis). Iškviečiamas `init()` metodas iš šios klasės tik tada, kai `isSupported()` gražina teigiamą reikšmę (`true`). `init()` metodas, naudodamas veiksmų architektūrą, gali pridėti grafinės sąsajos komponentus, ar atlikti kitas veiklas ar grąžinimus iš metodų.



12 pav. MagicDraw įskiepio panaudojimas (pagal (No Magic, 2007))

MagicDraw įskiepio kūrimas

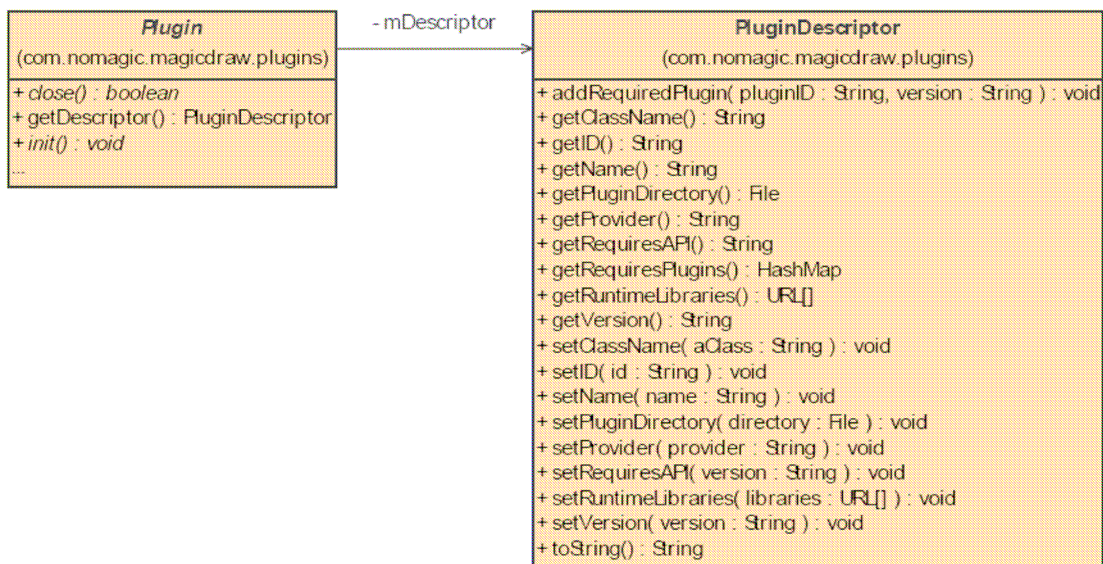
Įskiepi kuriant reikia atlikti šiuos veiksmus:

1. Sukurti plėtinio direktoriją „plugins“ direktorijoje;
2. Parašyti plėtinio kodą. Plėtinys turi turėti bent klasę, kuri paveldi `com.nomagic.magicdraw.plugins.Plugin` klasę
3. Sukompiliuoti ir supakuoti plėtinį į jar failą;

Kodo kompiliavimui turi būti įtrauktos klasės į sisteminius nustatymus `java classpath`. Sukompiliuotas kodas turi būti supakuotas į jar failą. Jar paketo formavimui reikia naudoti `jar` komandą „plugins“ direktorijoje (pvz. `jar -cf myplugin\myplugin.jar myplugin *.class`)

4. Parašyti plėtinio aprašymą; Plėtinio aprašymas turi būti faile `plugin.xml`. Šis failas turi būti patalpintas kuriamo plėtinio direktorijoje.

Rašant įskiepio kodą būtina panaudoti bent vieną apibrėžtą klasę iš `com.nomagic.magicdraw.plugins.Plugin` klasių.



13 pav. Įskiepio klasės (No Magic, 2007)

Plugin yra bazinė abstrakti klasė visiems MagicDraw įskiepiams (pav. 13). Kuriamas įskiepis turi būti išplėstinis iš šios klasės. Kiekvienas įskiepis turi savo deskriptorių nuskaitomą įskiepių tvarkytojo. Įskiepis turi tris specialius metodus:

- `public abstract void init()` metodas yra vykdomas kraunantis MagicDraw programai. Įskiepis turi perimti šį metodą ir realizuoti savo funkcionalumą.
- `public abstract boolean close()` metodas yra vykdomas išsijungiant MagicDraw programai. Įskiepis turi perimti šį metodą ir gražinti teigiamą reikšmę (*true*), jei įskiepis yra pasiruošęs išjungimui arba neigiamą reikšmę (*false*) kitu atveju ir programos išjungimas bus atšauktas.
- `public abstract boolean isSupported()` metodas yra vykdomas prieš įskiepio `init()` metodą. Įskiepis nebus įtraukiamas, metodui gražinant neigiamą reikšmę (*false*).

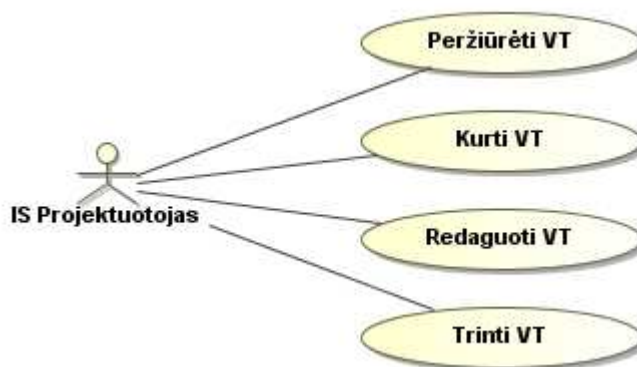
3. MagicDraw UML įskiepio Roso veiklos taisyklėms modeliuoti projektas

Projekto uždavinys – suprojektuoti ir realizuoti MagicDraw UML įskiepi, skirtą veiklos taisyklių braižymui ir jų susiejimui su klasių, atributų, asociacijų ir priklausomybių ryšio (dependency) elementais. Taip pat turi būti parengta atitinkama VT modeliavimo metodika.

Kuriamos sistemos projektavimui bus naudojamas MagicDraw UML paketas, kuris jau eilę metų yra pripažįstamas vienu geriausių projektavimo įrankių.

3.1. Funkciniai reikalavimai

MagicDraw UML įskiepis leis IS projektuotojui valdyti (pav. 14) veiklos taisyklės panaudojant esamų modelių elementus. Jis galės peržiūrėti esamas sukurtas VT projekte, kurti naujas VT, kuriant panaudoti esamas VT, redaguoti esamas VT, bei trinti sukurtas VT.



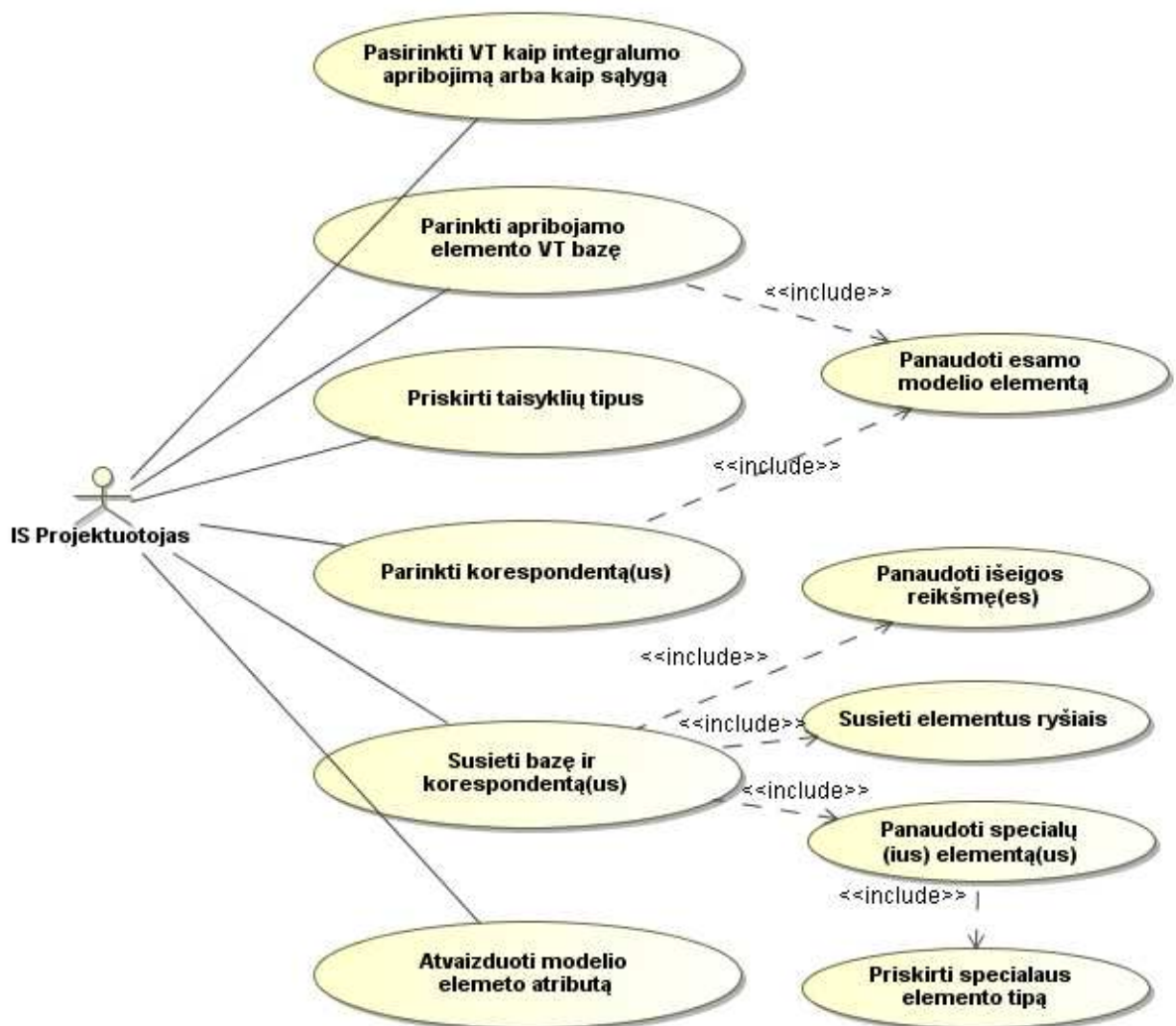
14 pav. Veiklos taisyklės valdymas

Duomenų modelio elementus galima įkelti iš modelio medžio esančio standartiniu komponentu MagicDraw UML įrankyje. Taip pat brėžiami VT elementai bus saugojami modelyje.

Kurti VT atvejis turi visas funkcijas, kurios yra svarbiausios VT modeliavimui (pav. 15):

- Pasirinkti VT kaip integralumo apribojimą arba kaip sąlygą. Galimybė pasirinkti iš meniu punkto integralumo apribojimą arba kaip sąlygą.
- Parinkti apribojamo elemento VT bazę. Pagrindinis tikslas - galimybė panaudoti esamo modelio elementus, taip pat panaudoti jau esamas VT ar jų komponentus.
- Priskirti taisyklių tipus. Galimybė pasirinkti vieną iš visu Roso notacijoje apibrėžtų nedalomų ir sudėtinių taisyklės tipų.

- Parinkti korespondentą(us). Pagrindinis tikslas - galimybė panaudoti esamo modelio elementus, taip pat panaudoti jau esamas VT ar jų komponentus.
- Susieti bazę ir korespondentą(us). Ši funkcija susideda iš trijų dalių: Panaudoti išeiigos reikšmę, kuri gali būti ir kitos taisyklės; Susieti visus taisyklėje esančius elementus atitinkamais ryšiais (pagal taisyklės reikšmę); Panaudoti specialų simbolį arba keletą jų ir jam parinkti tipą iš visu Roso notacijoje apibrėžtų tipų (žr. Priedas nr.1);
- Atvaizduoti modelio elemento atributą. Išskirti modelio elemento atributą kaip atskirą elementą, tam kad būtų galima grafiškai išreikšti taisyklei priklausančius elementus.



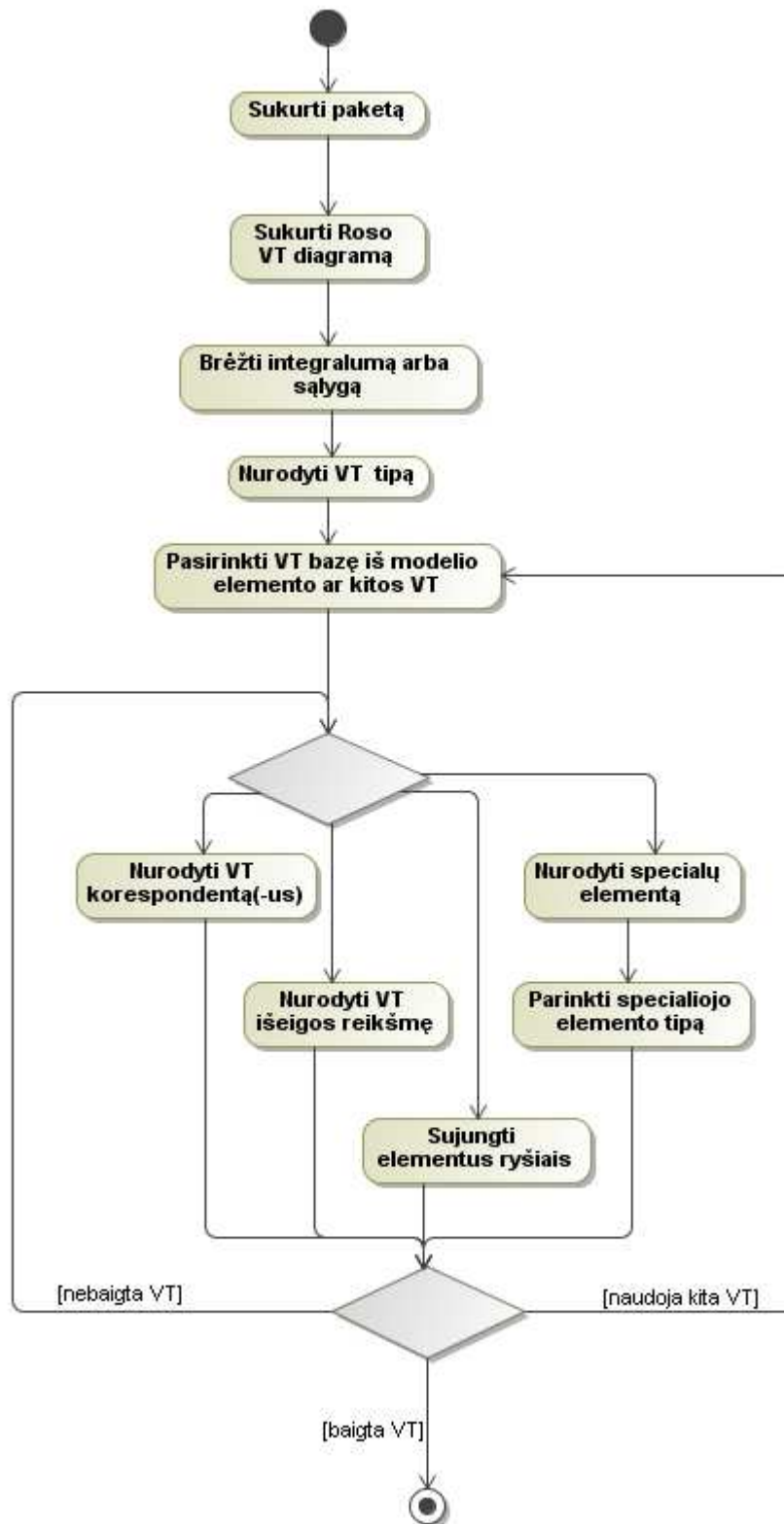
15 pav. Veiklos taisyklės kūrimo funkcijos

Modelio elemento parinkimo panaudojimo atvejis yra realizuotas MagicDraw UML produkte, todėl jį reiks panaudoti.

3.2.VT modeliavimo metodika

VT formalus aprašas prasideda nuo VT žodinio aprašo. Tai padeda lengviau braižyti taisyklę diagramoje. Siūlomas taisyklės kūrimo eiliškumo procesas pavaizduotas veiklos diagramoje (pav. 16).

- VT reikia pradėti nuo taisyklės išsirinkimo, t.y. pasirinkti ir brėžti integralumo arba sąlygos simbolį.
- Veikos taisyklės tipą galima pasirinkti iš profilyje įvestų VT tipų, kurie yra apibrėžti Roso notacijoje. Parinkti taisyklės tipą (arba nustatyti, kokius testus ji atlieka) yra būtina, nes tik esant nurodytam tipui galima teisingai interpretuoti taisyklę.
- Modelio elementui yra parenkamas stereotipas „BRule Anchor“ (inkaras), taip jis priskiriamas VT bazei. Esant šiam elementui klasės diagramoje bus galima atsidaryti VT diagramą, kurioje šis elementas yra tos VT bazė. Taip pat bazė gali būti kita VT ar jos išeišos reikšmė.
- Korespondentą nurodyti galima brėžiant ryšį iš VT į kitą VT ar jos išeišos reikšmę ar modelio elementą (klasę, atributą, asociacijos ar priklausomybės ryšį).
- Paprastai taisyklės išeišos reikšmė yra “nematoma” ir vartojama taisyklės viduje tam tikram testavimui atlikti arba apribojimui pritaikyti. Tačiau tam tikrose taisyklėse reikia tiesiogiai tikrinti kitų taisyklių išeišos. Tuomet VT išeišos reikšmė atsispindės ir taisyklės, vartojančios šią reikšmę, diagramoje.
- Galimybė naudoti papildomus simbolius, galinčius pakeisti standartinę tam tikro tipo taisyklės interpretaciją. Šie simboliai skirstomi į dvi pagrindines grupes:
 1. Interpretatoriai – tai simboliai, nurodantys, jog taisyklę jos bazės atžvilgiu reikia interpretuoti ne taip, kaip reikėtų remiantis tik jos tipu. Šie simboliai yra žymimi ant bazės ryšio rodyklės.
 2. Apribojimai – tai simboliai, nurodantys, jog taisyklei jos korespondento(u) egzempliorių atžvilgiu turi būti taikomi specialūs apribojimai.
- Jiems būtiną nurodyti tipą pasirenkant iš profilyje esamų specialiųjų simbolių tipų. Specialiųjų simbolių braižymo semantika turi tenkinti priede (Priedas nr.1) nurodytas taisykles.
- Susieti elementus ryšiais: tiek išeinantis iš bazės ryšys, tiek ir einantys į korespondentus ryšiai VT diagramoje visada vaizduojami punktyrine rodykle. Ryšys tarp VT ir jo išeišos reikšmės vaizduojamas vientisa linija.
- VT turi būti papildoma, tol kol bus visiškai baigta.



16 pav. VT kūrimo veiklos diagrama

3.3. Vartotojo sąsajos reikalavimai

VT braižymo režimo inicijavimas

VT galima braižyti pasirinkus Roso veiklos taisyklių diagramą su aktyvuotu Roso veiklos taisyklių profiliu (angl. profile).

VT braižymas

VT braižymo semantika turi tenkinti 2.3 skyriuje aptartą Roso metodą.

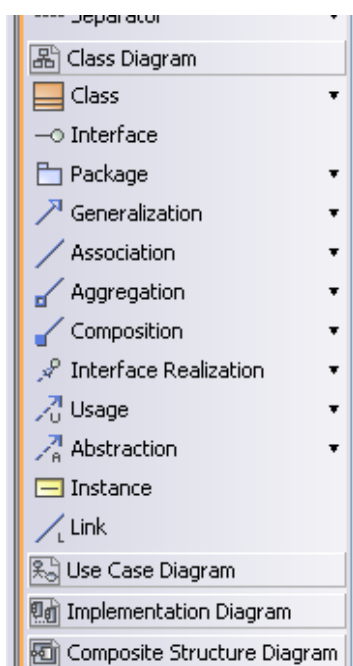
Veiklos taisyklių įvedimas ir redagavimas turi būti realizuotas taip, kad būtų tenkinami visi formalūs baziniai reikalavimai pateikti aukščiau.

VT elementų pasirinkimas ir braižymo apribojimai

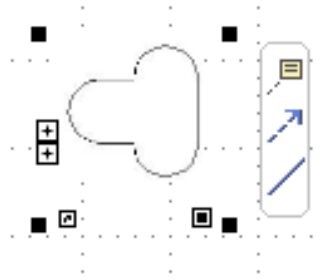
Taisyklės elementų pasirinkimas bus galimas iš trijų tipų meniu:

- Diagramos elementų pasirinkimo meniu (pav. 17) bus sudarytas iš trijų meniu punktų rinkinio: bendrų (angl. common) elementų rinkinio, klasės diagramos elementų rinkinio ir Roso VT diagramos elementų rinkinio.
- Pasirinktam diagramoje elementui pasirodžiusiame greitajame meniu (pav. 18) galima pasirinkti iš šio elemento leidžiamus brėžti elementus.
- Pasirinkus ryšio elementą, brėžti turi būti leidžiama tik tarp jam skirtu elementų (žr 3.5 skyrių).
- Brėžiant ryšį į adresatą (angl. target), galima pasirinkti naują siūlomą adresatą iš papildomo meniu (pav. 19).

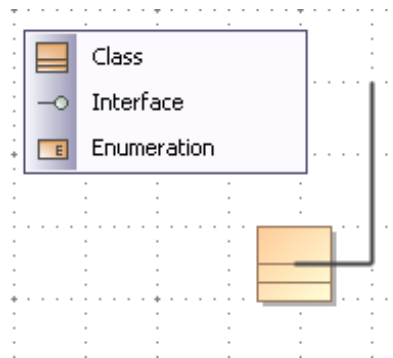
Visi braižymo elementai yra sukuriami kuriant profilį.



17 pav. Diagramos elementų pasirinkimo meniu pavyzdys



18 pav. Diagramos elemento greitojo meniu pavyzdys



19 pav. Brėžiamo ryšio galimo naujo adresato papildomo pasirinkimo meniu pavyzdys

3.4. Nefunkciniai reikalavimai

Pagrindinis reikalavimas išlaikyti braižymo paprastumą kaip yra realizuotas elementų braižymas MagicDraw UML. Elementai ir jų pasirinkimas turi būti intuityviai suprantami.

Prieš kuriant įskiepį reikia apsibrėžti jo nefunkcinius reikalavimus. Kad būtų apibrėžtos jo kokybės ir santykio su išoriniais veiksniais gairės.

Taigi kuriamas produktas turi:

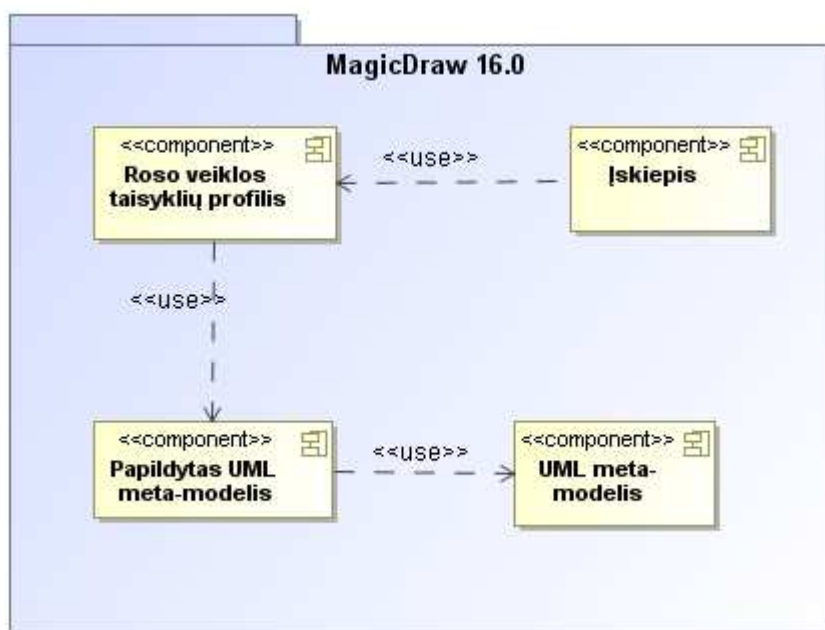
- išlaikyti braižymo paprastumą kaip yra realizuotas elementų braižymas MagicDraw UML prodykte.
- elementai ir jų pasirinkimas turi būti intuityviai suprantami.
- nesugadinti sistemos į kurią įskiepis yra integruojamas;
- būti nesudėtingai naudojamas;
- lengvai iškviečiamas;
- turėti galimybę būti paleidžiamas kelis kartus;
- būti lengvai pašalinamas.

3.5.Sistemos projektas

MagicDraw UML yra sukurta Java technologijos pagrindu, tai bus naudojamas laisvai pasirenkamas Java kalbos programavimo sąsajos paketas Eclipse.

Veiklos taisyklių saugykla yra MagicDraw UML naudojamas modelio elementų išsaugojimas xml struktūroje (No Magic, 2007).

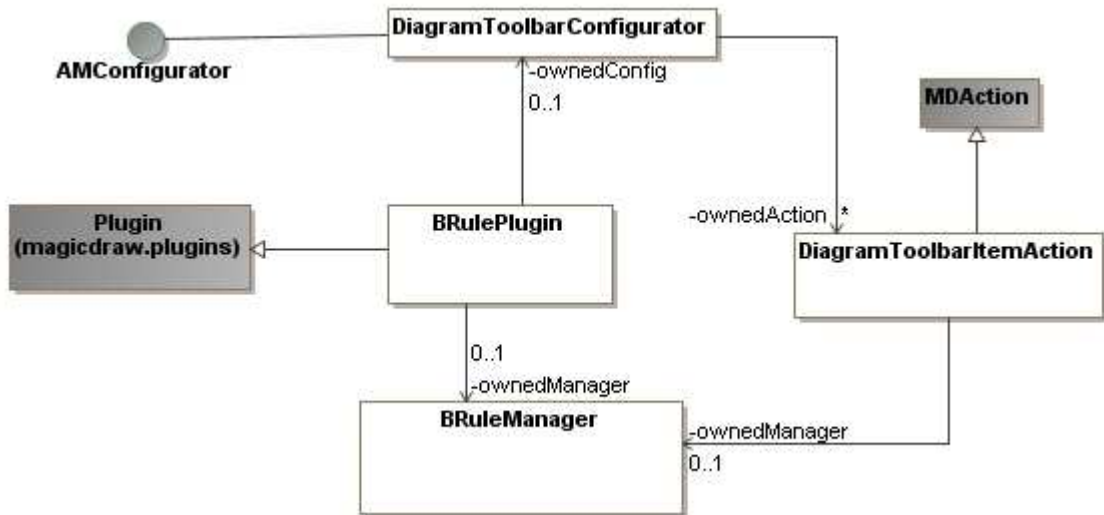
Kuriamo sprendimo pagrindiniai komponentai pateikti sekančiai (pav. 20):



20 pav. Sistemos komponentų diagrama

Kuriamas įskiepis naudoja Rosos veiklos taisyklių profilį, kuris saugo VT tipus, specialiųjų elementų tipus. Šis profilis papildo MagicDraw UML produkte realizuotą UML meta-modelį stereotipizuotais elementais paveldėtai iš šio modelio.

MagicDraw įskiepis (klasių diagrama pav. 21) yra valdymo elementas tarp profiliu realizuojamo VT braižymo ir MagicDraw UML esamų (kuriamų) modelio elementų.



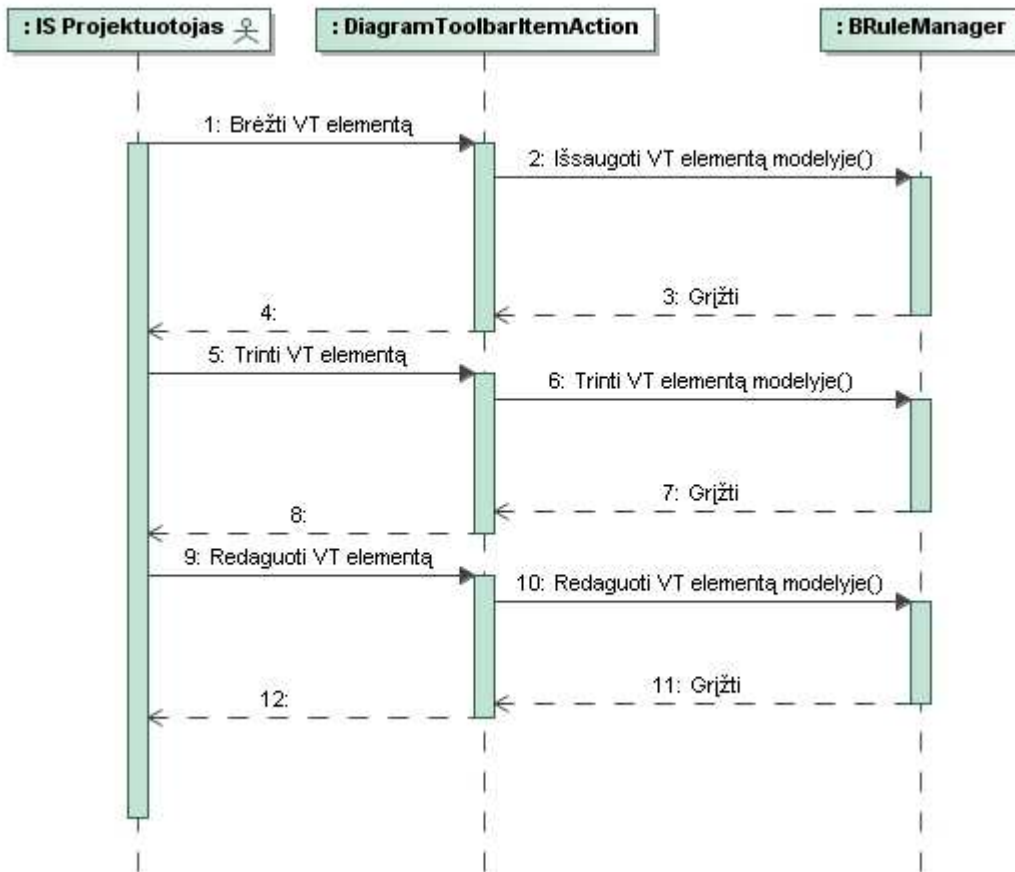
21 pav. Įskiepio klasių diagrama (pilka spalva išskirtos klasės pavaldėtos iš MagicDraw UML įrankio)

BRulePlugin (VT įskiepio) įskiepio užkrovimo klasė, kurios init() metodas yra inicijuojamas startuojant MagicDraw UML produktą. DiagramToolbarConfigurator (diagramos įrankių juostos konfiguratorius) klasė naudojama sukurti mygtukų veiksmams. DiagramToolBarItemAction (diagramos įrankių juostos mygtukų veiksmų) klasė perdengs su profiliu sukurtų stereotipizuotų elementų sukūrimo veiksmus, tam kad būtų galima įrašyti kuriamus VT elementus į MagicDraw UML elementų saugojimo struktūrą (XML struktūrą). BRuleManager (VT valdiklio) klasė yra pagrindinė įskiepio valdymo klasė. Skirta užkrauti Roso veiklos taisyklių profilį, įvesti papildomą Roso diagramą ir jos pasirinkimo meniu punktą. Tikrinti interpretatorių ir apribojimų parinkimą (žr. Priedas nr.2)

Pažymėtina, jog praplečiant modelį buvo naudojamos sąvokos anglų kalba. Taip siekiama užtikrinti viso modelio vientisumą bei sukurti tvirtesnes prielaidas praplėtimo taikymui praktikoje.

3.5.1. Sistemos elgsenos modelis

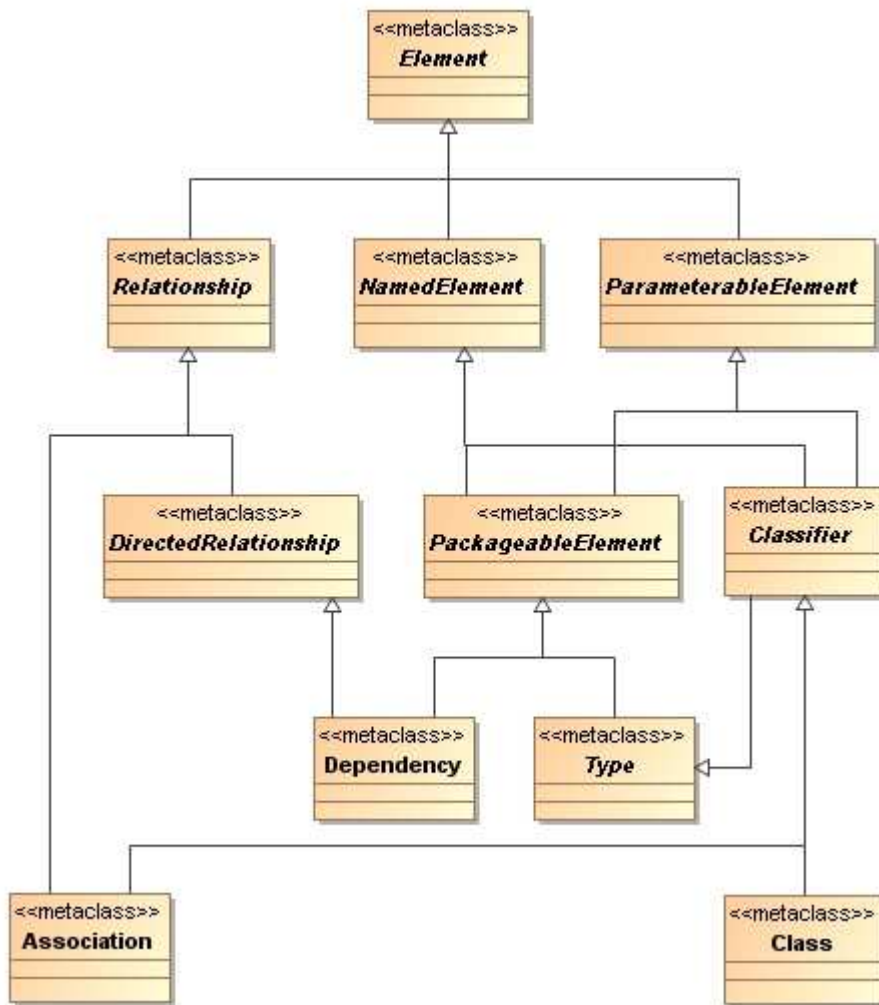
MagicDraw vartotojas (IS Projektuotojas) brėždamas, redaguodamas, trindamas VT elementą iššaukia veiksmus, kuriuos įskiepis perdengia ir vykdo veiksmus, kad įrašytų, redaguotų, trintų VT esančius modelyje (pav. 22).



22 pav. MagicDraw UML vartotojo ir VT įskiepio elgsena

3.5.2. UML metamodelis ir jo praplėtimo aprašas

MagicDraw UML glaudai bendradarbiauja su OMG. Buvo išsiaiškinta, kad MagicDraw UML pilnai palaiko OMG pateiktą UML2 specifikaciją. MagicDraw UML projektiniai metamodeliai yra realizuoti pagal OMG specifikacijas ir pilnai palaiko standartus. Visi atributai yra įgyvendinti iš UML specifikacijos (No Magic, 2007). UML specifikacijoje aprašyto UML metamodelio dalis (Object Management Group, 2007) pavaizduota pav. 23.



23 pav. MagicDraw UML programoje realizuoto UML meta-modelio dalis

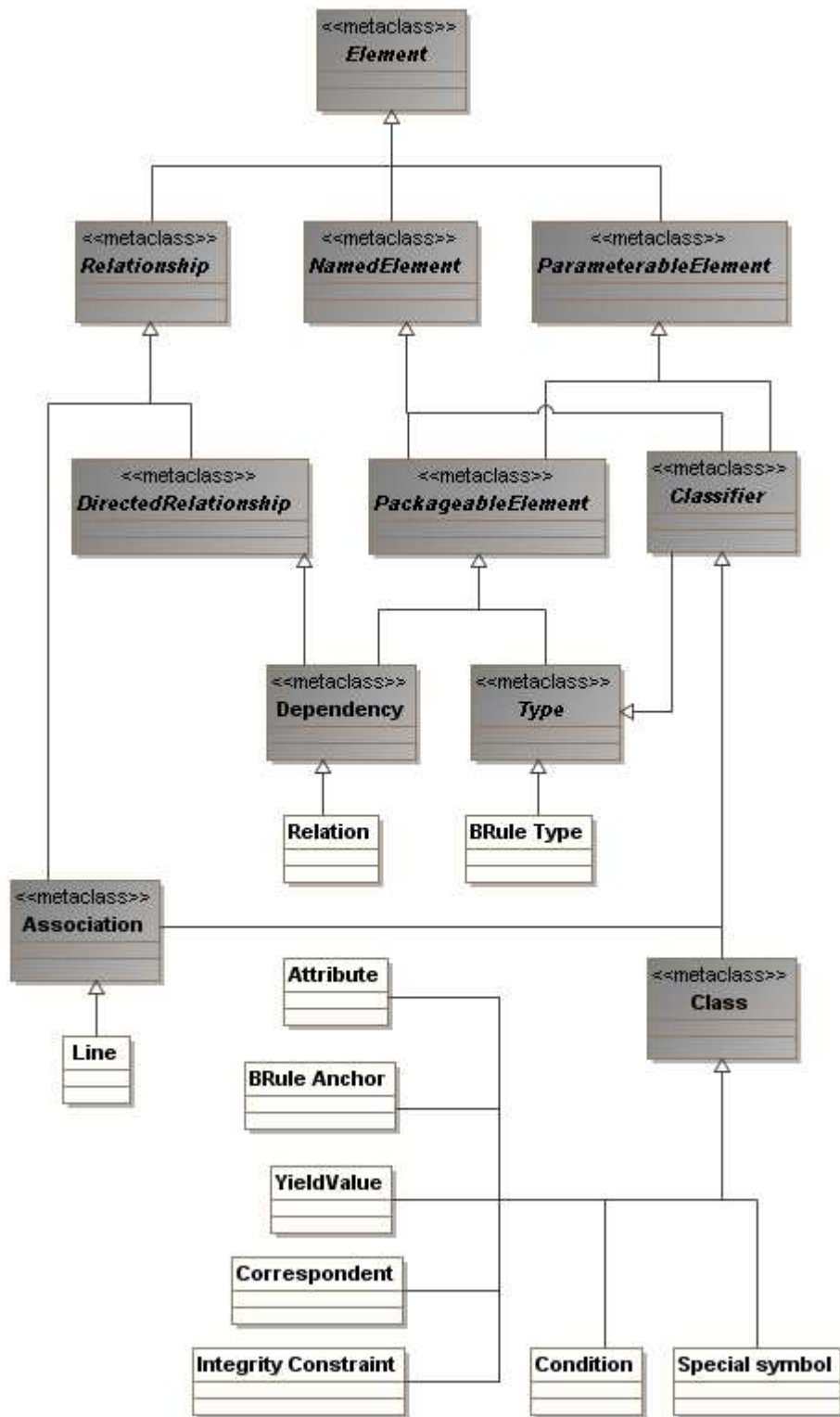
Pasiūlytas papildytas UML metamodelis (pav. 24). Kiekvienas VT naudojamas elementas yra paveldėtas iš UML meta-elemento. Paveldėtos klasės patektos lentelėje 1.

1 lentelė. VT elementų paveldėjimas iš UML meta-elementų

Elementas	Paveldėtas iš Meta-elemento	Aprašas
VT sąlyga (Condition)	Klasė (Class)	Elementas skirtas atvaizduoti VT kaip sąlygą.
VT apribojimas (Integrity Constraint)	Klasė (Class)	Elementas skirtas atvaizduoti VT kaip integralumo apribojimą.
VT tipas (BRule Type)	Tipas (Type)	Viena iš nedalomų arba išvestinių VT tipų.
Inkaras	Klasė (Class)	Žymi modelio elementą kaip VT bazę.

Elementas	Paveldėtas iš Meta- elemento	Aprašas
(BRule Anchor)		
Korespondentas (Correspondent)	Klasė (Class)	Žymi modelio elementą kaip VT korespondentą.
Atributas (Attribute)	Klasė (Class)	Skirta atvaizduoti modelio elementą - atributą kaip atskirą elementą.
Išėigos reikšmė (YieldValue)	Klasė (Class)	Skirta atvaizduoti VT galimą išėigos reikšmę, kuri gali būti kitos VT bazė arba korespondentas
Specialus simbolis (Special symbol)	Klasė (Class)	Vienas iš Roso notacijoje aprašytų specialiųjų simbolių saugomų profilyje
Linija (Line)	Asociacija (Asociacion)	Skirtas sujungti VT ir jos išėigos reikšmę
Ryšys (Relation)	Priklausomybė (Dependency)	Ryšys, kuris skirtas sujungti VT su baze ir korespondentu

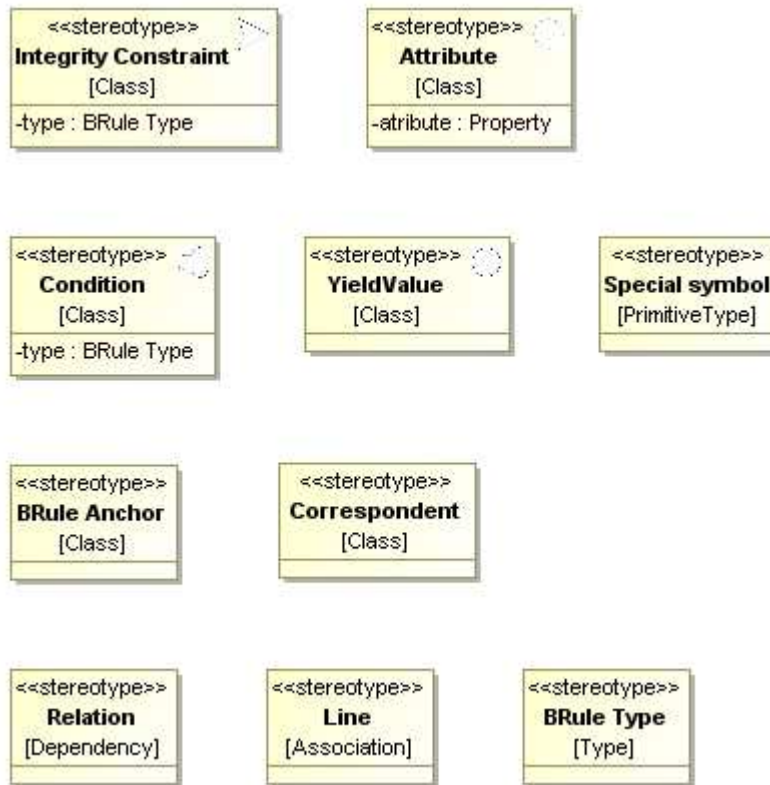
Pažymėtina, jog praplečiant modelį buvo naudojamos sąvokos anglų kalba. Taip siekiama užtikrinti viso modelio vientisumą bei sukurti tvirtesnes prielaidas praplėtimo taikymui praktikoje.



24 pav. MagicDraw UML programoje realizuoto UML meta-modelio dalis papildyta VT elementais (standartiniai elementai išskirti pilka spalva, naujieji - balta)

3.5.3. Roso veiklos taisyklių profilis

Lentelėje 1 aprašyti elementai yra sukurti kaip stereotipai (pav. 25).



25 pav. Roso veiklos taisyklių profilio VT vaizdavimo elementai

Visi VT elementai (lentelė 1) yra detalizuojami profilyje.

- Taisyklės sąlyga ir integralumo apribojimas turi atributą tipas (`type`), kurio tipas yra VT tipas. Elementų tipą reikia nurodyti elemento specifikacijos lange (žr pav. 29). Jie taip pat turi savo vaizdavimo simbolius (žr pav. 3). Braižant šiuos grafinius simbolius būtina užtikrinti jų galiojimo kryptį.
- VT tipai yra suvesti Roso veiklos taisyklių profilyje. Pagal Roso notaciją yra 32 nedalomų ir 58 išvestinių taisyklių tipai (Ross, 1997).
- Inkaras stereotipas yra skirtas priskirti modelio elementui – klasei galimybę būti panaudotam kaip VT bazė. Šios klasės greitis meniu papildomas ryšys (`Relation`) bei linija (`Line`) elementų pasirinkimo punktais.
- Korespondentas stereotipas yra skirtas priskirti modelio elementui – klasei galimybę būti panaudotam kaip VT korespondentas.
- Atributas skirtas atvaizduoti grafiškai modelio elemento (pvz.: klasės, paketo ir t.t.) atributą. Jis turi atributą `attribute`, kurio tipas yra savybė (angl. `Property`). Atvaizduojamą atributą reikia nurodyti grafinio elemento specifikacijos lange. Šio grafinio elemento simbolis yra apskritimas.

- Išeigos reikšmė skirta tam tikram testavimui atlikti arba apribojimui pritaikyti. Tam tikrose taisyklėse reikia tiesiogiai tikrinti kitų taisyklių išeigos reikšmes (tai labiau būdinga sąlygos tipo taisyklėms). (Ross, 1997).
- Specialus simbolis skirtas pakeisti standartinę tam tikro tipo taisyklės interpretaciją. Šie simboliai skirstomi į dvi pagrindines grupes:
 - Interpretatoriai – tai simboliai, nurodantys, jog taisyklę jos bazės atžvilgiu reikia interpretuoti ne taip, kaip reikėtų remiantis tik jos tipu. Šie simboliai yra brėžiami tarp bazės ir taisyklės.
 - Apribojimai – tai simboliai, nurodantys, jog taisyklei jos korespondento(-ų) egzempliorių atžvilgiu turi būti taikomi specialūs apribojimai (Butleris; Kapočius, 2002).

Visi simboliai yra įvesti profilyje, o išsamus jų sąrašas pateikiamas šio darbo 1 priede. Svarbus aspektas, kad simbolių teisingas panaudojimas interpretatoriumi ar apribojimu būtų tikrinamas pagal Roso notacijoje aprašytus kriterijus (žr. Priedas nr. 2).

Specialus simbolis yra brėžiamas diagramoje ir parenkamas per papildomą elemento meniu stereotipo parinkimo langą. Tada pasikeis elemento grafinis simbolis.

- Linija skirta sujungti elementą ir jo atvaizduojamą atributą atskiru grafiniu elementu. Taip pat skirta sujungti VT ir jos išeigos reikšmę.
- Ryšys skirtas sujungti elementus, kurio ištaka (angl. source) vienas iš: VT sąlyga, VT integralumo apribojimas, Inkaras, Atributas, Išeigos reikšmė, Specialus simbolis; o adresatas (angl. target) vienas iš: VT sąlyga, VT integralumo apribojimas, Korespondentas, Atributas, Specialus simbolis.

Profilis yra užkraunamas kartu su VT įskiepiu.

4. Sukurtojo įskiepio eksperimentinis įvertinimas

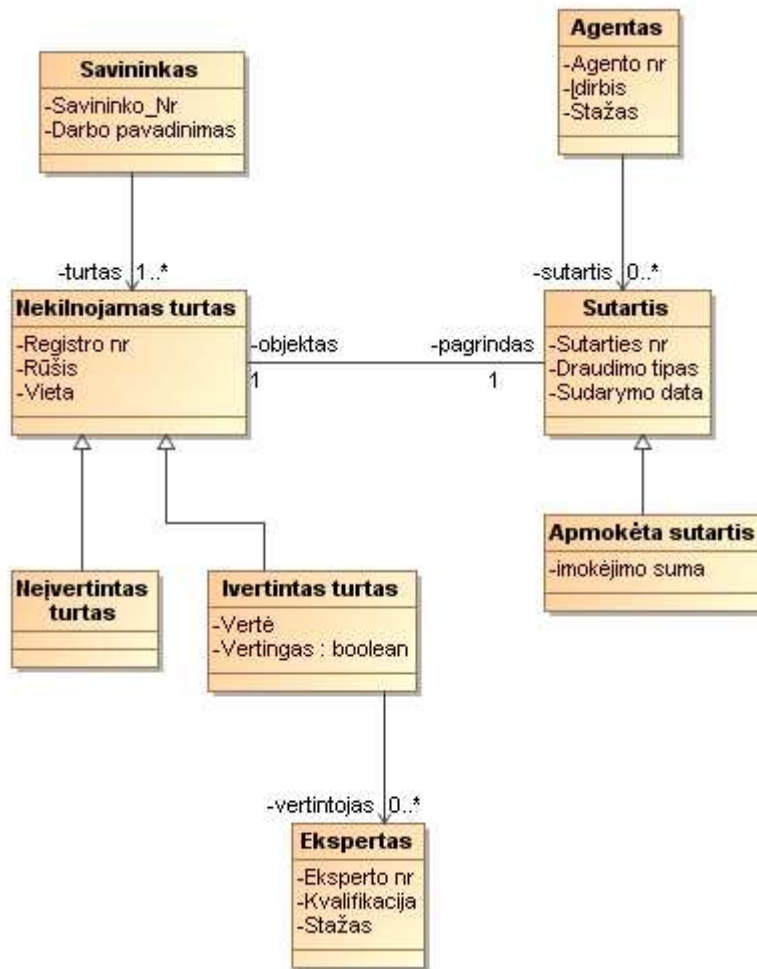
4.1. Eksperimento koncepcija

Eksperimentui buvo nuspręsta pasirinkti supaprastintą nekilnojamo turto draudimo procesą. Pagrindinė dalykinės srities paskirtis – inicijuoti įvairaus pobūdžio VT, leidžiančias įvairiapusiškai ištestuoti sukurtą pagal Roso metodo notaciją VT braižymo ir susiejimo su modelio elementais įskiepi.

Nagrinėjama apibendrinta nekilnojamo turto draudimo sistema. Paprastumo dėlei draudimo rūšis nekonkretizuojama, analizuojamas universalus atvejis. Norintis drausti nekilnojamą turtą asmuo (toliau savininkas) bendrauja su draudimo bendrovės atstovu (toliau agentas). Savininkas kreipiasi į agentą su raštišku prašymu užregistruoti jo nekilnojamą turtą. Agentas prašymo pagrindu pateikia užklausą bendrovės turto vertinimo ekspertui (toliau ekspertas). Ekspertas agentui pateikia nekilnojamo turto įvertinimą. Jeigu ekspertas nustato, jog nekilnojamas turtas nepakankamai vertingas, agentas savininkui praneša, kad draudimo prašymas yra atmestas. Ekspertui nustačius, jog nekilnojamas turtas yra pakankamai vertingas, agentas savininkui pateikia siūlomą draudimo sutartį. Siekiant pernelyg nekomplikuoti užduoties, priimama, jog agentas pats pasiūlo draudimo tipą ir kitas sąlygas. Savininkas, įvertinęs sutartį, ją arba atmeta, arba priima. Nutaręs priimti sutartį, savininkas tiesiog įmoka joje numatytą pinigų sumą, o agentas, gavęs pinigus, sutartį registruoja ir išduoda savininkui draudimo polisą. Sutartį agentas gali siūlyti tol, kol savininkas ją priims. Nutaręs, jog savininkui įtikti nepavyks, agentas gali atmesti prašymą.

4.2. Dalykinės srities duomenų modelio sudarymas

Pagal pateiktą dalykinės srities aprašymą buvo sudarytas duomenų modelis. Atsižvelgiant į pasiūlyme suformuluotus reikalavimus, tokiu modeliu buvo klasių diagrama (26 pav.).



26 pav. Apibendrinta nekilnojamo turto draudimo sistema

4.3. Veiklos taisyklių specifikavimas

Pagal pasirinktą dalykinę sritį buvo sudarytas VT teiginių rinkinys. Išskiriant testines taisykles nebuvo siekiama išsamiai ir tiksliai specifikuoti dalykinę sritį ar jos kritines specifikavimo dalis. Pagrindinis tikslas buvo parinkti nedaug skirtingo pobūdžio VT, kurios atspindėtų kuo įvairesnes struktūrines kombinacijas. Vienas iš eksperimento tikslų yra palyginti sukurtąjį VT modeliavimo pagal Roso metodą prototipą su analizės metu ištirtu Visio įrankiu (žr. šio darbo 2.2.1 skyrių). Dėl šios priežasties sudarant bandomąjį VT rinkinį numatoma remtis minėto Visio įrankio testavimui naudotų taisyklių struktūra (Kapočius, 2006)]. Tokiu būdu bus galima objektyviau įvertinti abiejų įrankių pajėgumus.

Buvo suformuluoti šie VT teiginiai:

1. Eksperto, kuris įvertino 20 ar mažiau nuosavybių, kvalifikacija privalo nekisti.
2. Sutartyje būtina nurodyti jos sudarymo datą ir ją sudariusį agentą.
3. Agento įdirbis privalo būti lygus apmokėtų sutarčių, sudarytų šio agento, bendrai įmokėjimo sumų sumai.

4. Eksperto stažas privalo būti didesnis arba lygus 2 metams.
5. Jeigu nekilnojamasis turtas yra įvertintas, tačiau nėra klasifikuojamas kaip pakankamai vertingas ir pagal jį nesudaryta sutartis, tai jis privalo būti klasifikuojamas kaip pakankamai vertingas prieš tai, kai pagal jį galės būti sudaryta sutartis.
6. Įvedant naują arba modifikuojant esamą visų ekspertų stažą, jis privalo didėti.
7. Nekilnojamasis turtas negali būti įtrauktas į daugiau nei 5 sutartis.

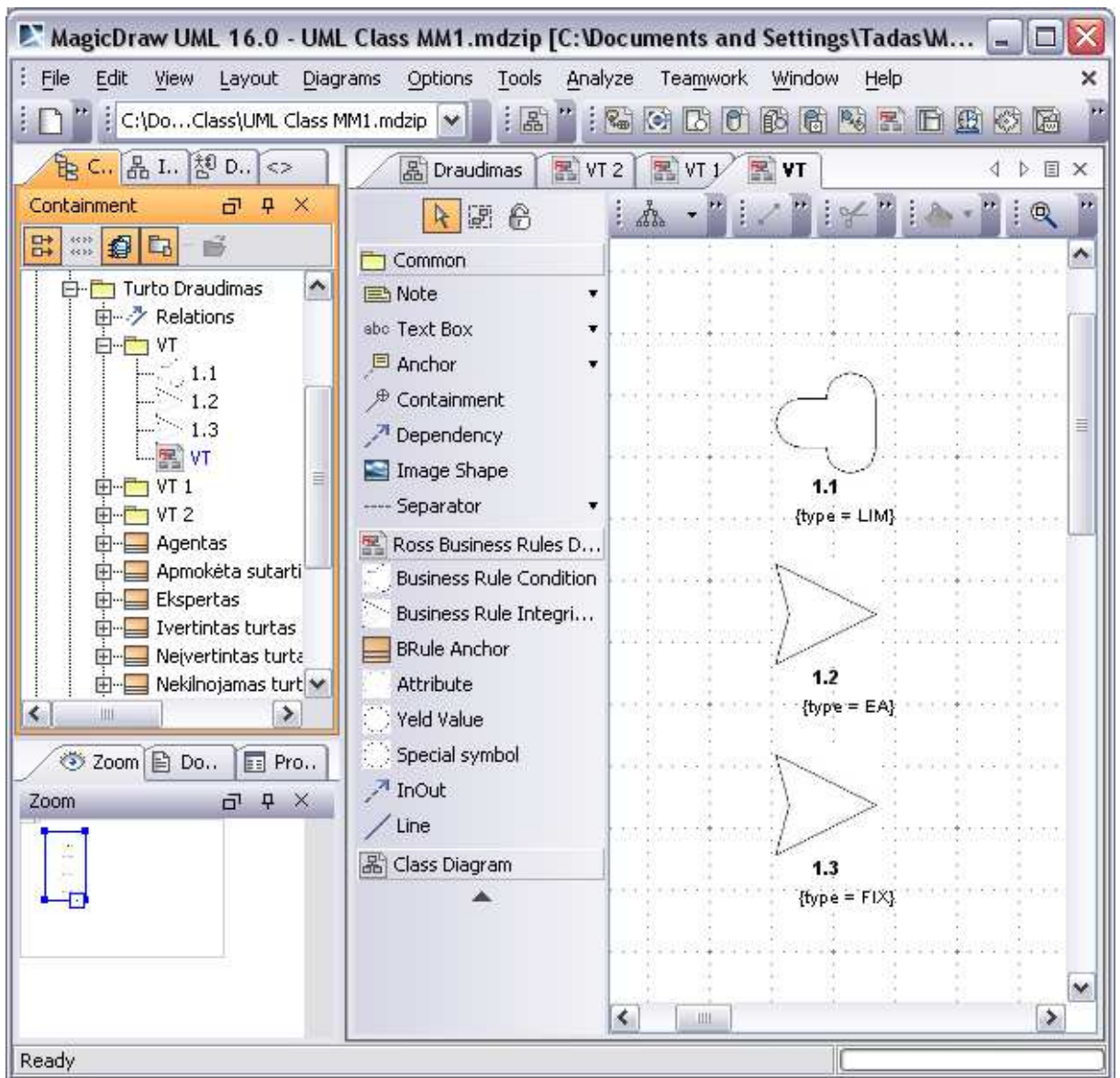
Vadovaujantis duomenų modeliu bei suformuluotais VT teiginiais buvo kuriamos taisyklių diagramos. VT modeliavimas buvo atliekamas naudojantis sukurtuoju MagicDraw UML įskiepiu.

4.4.Pasirengimas eksperimentui

Įskiepi sudaro vienas failas Ross_Business_Rules_Plugin_v1.0.zip.

- Įskiepio diegimo langas paleidžiamas pasirinkus MagicDraw UML meniu Help grupės Resource/Plugin Manager punktą.
- Atsidariusiame lange paspaudus Import mygtuką ir atsidarius failo parinkimo dialogui nurodyti įskiepio failą.

Įrankio darbo lango pavyzdys su parinkta Roso veiklos taisyklių diagrama pateiktas 27 pav.



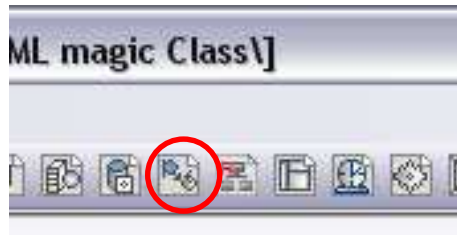
27 pav. MagicDraw UML su Roso veiklos taisyklių įskiepiu langas

4.5. Eksperimento su sukurtuoju įskiepiu VT kūrimo eiga

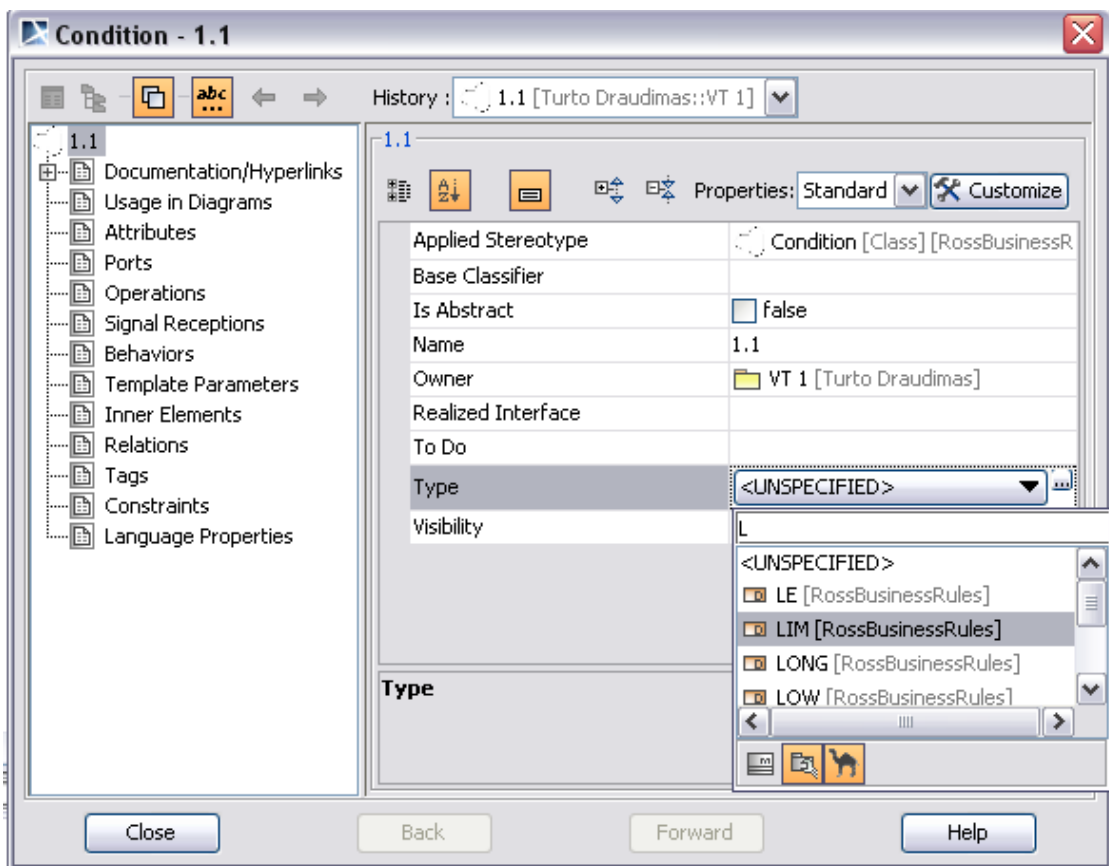
VT braižymą vykdysime pagal apibrėžtą VT modeliavimo pasiūlytą eigą (žr. šio darbo 3.1.1 skyrių).

- Pradedame susikurdami katalogą VT1 ir sukuriame Roso VT diagramą pasirenkant mygtuką iš įrankių juostos (28 pav. apibrėžtas raudonai).
- Užrašome pirmos suformuluotos VT teiginį į diagramos braižymo langą, taip bus paprasčiau interpretuoti VT. VT teiginiui (*Eksperto, kuris įvertino 20 ar mažiau nuosavybių, kvalifikacija privalo nekisti*) parenkame ir brėžiame du integralumo apribojimo ir vieną sąlygos elementus (27 pav.). Brėžimo elementus pasirenkame iš diagramos meniu (27 pav. kairiau nuo diagramos braižymo lauko).

- Kiekvienos veiklos taisyklės specifikacijų lange (29 pav.) parenkame VT tipą.



28 pav. MagicDraw UML su Roso VT įskiepiu Roso VT diagramos sukūrimo mygtukas

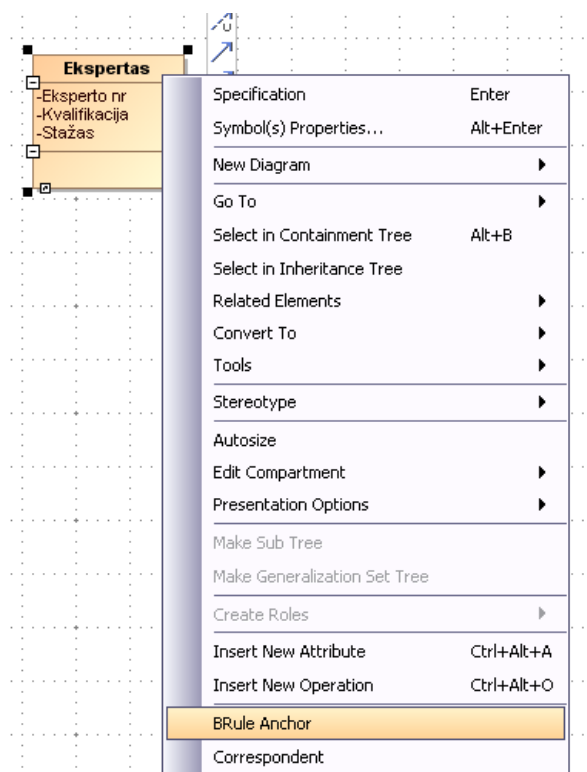


29 pav. VT specifikacijų langas su savo parametru tipas (Type)

Šiuo atveju sąlygai – apribota (LIM), integralumo apribojimams – įgalinta (EA) ir išaldyti (FIX) (27 pav.).

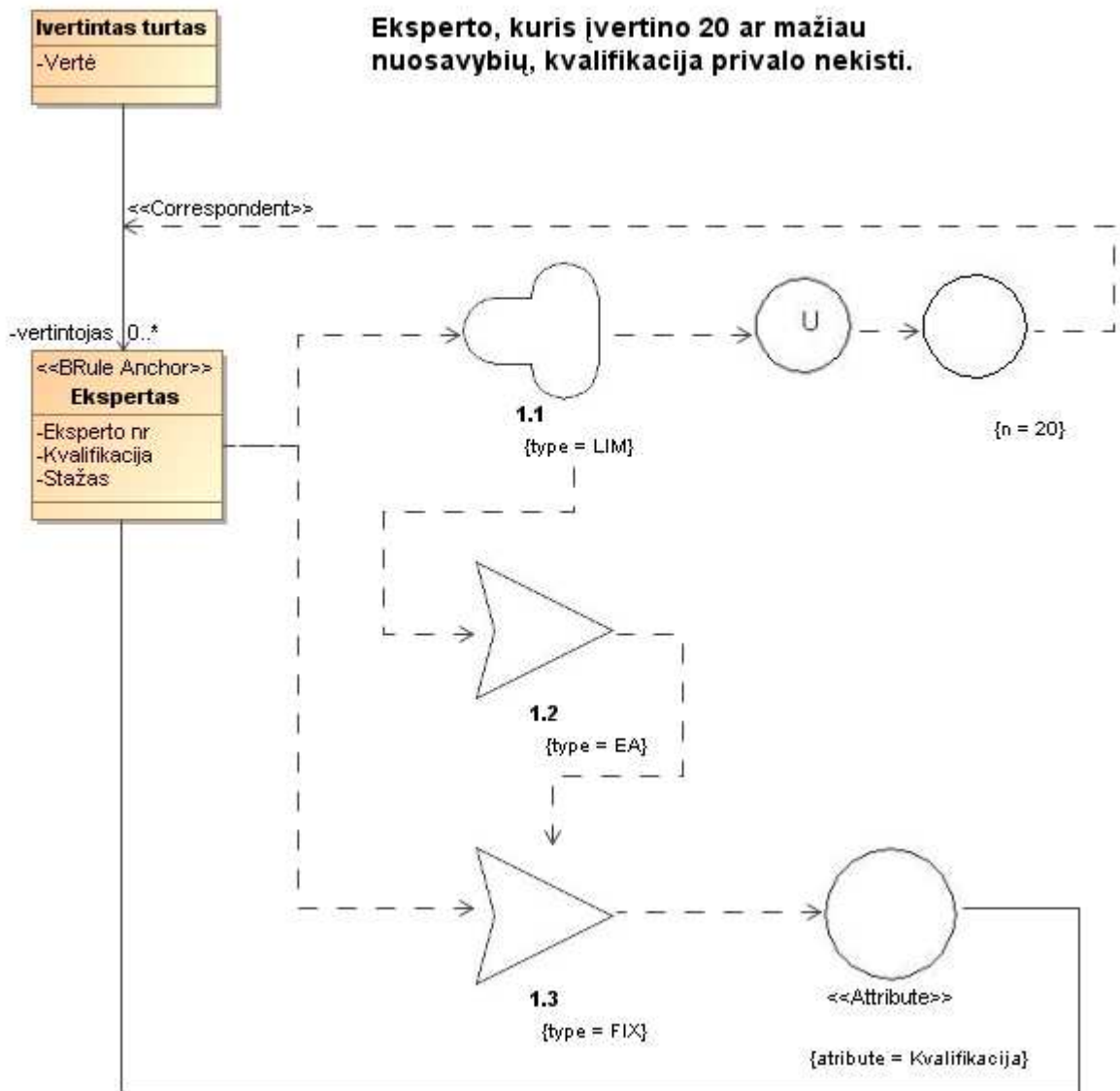
- Svarbiausias aspektas yra pasirinkti arba kurti modelio elementus ir susieti su taisykle. Pasirenkame Ekspertas klasę iš Turto Draudimo modelio klasių diagramos ir iškeliamė į Roso VT diagramą VT1. Parenkame įkeltai klasei stereotipą *BRule Anchor* (30 pav.). Taip modelio elementas Ekspertas tampa inkaru veiklos taisyklėje (VT baze) ir bus galima matyti kitose diagramose, kad jis yra naudojamas taisyklėje.

Nunaviguoti į kitą diagramą, kur šis elementas yra panaudotas, galima per elemento papildomo meniu punktą „eiti į“ (Go To).



30 pav. Papildomas elemento meniu su pažymėtu BRule Anchor punktu

- Brėžiame du specialiuosius simbolius (Special Symbol) ir parenkam jiems stereotipus specialus apibrėžiantis simbolis (mU) ir konstanta (n) (31 pav.). *Ekspertas* elemento *Kvalifikacija* atributą atvaizduojam kaip atskira simbolį (31 pav. skritulys su stereotipu <<Attribute>>) jis yra 1.3 VT korespondentas. 1.2 VT bazę brėžiame iš 1.1 VT išeigos, o korespondentą į 1.3 VT įeigą.
- Įtraukiame papildoma klasę *Ivertintas turtas* ir asociaciją tarp šios klasės ir *Ekspertas* klasės iš *Turto Draudimas* modelio. Minėtai asociacijai parenkame *Correspondent* stereotipą ir ją nurodome 1.1 VT korespondentu sujungiant ryšiu.
- Veiklos taisyklė yra pilna ir baigta (31 pav.).



31 pav. Pilna VT

Atlikus VT braižymą galima teigti, kad sukurtas įskiepis yra pažangesnis daugeliu aspektų už aptartą įrankį (žr. šio darbo 2.2.1 skyrių). Skiriasi esminiai dalykai:

- VT diagrama taip kaip užsaugoma, taip ir atkuriamą užkraunant;
- susiejama VT su konkrečiu standartizuotu modeliu (UML klasių diagrama);
- grafinių elementų pasirinkimas ir panaudojimas intuityviai suprantami;
- sudarytos VT kūrimo metodikos (žr. šio darbo 3.2 skyrių) taikymas ženkliai sutrumpina modeliavimo laiką;
- atliekama sintaksinio ir, iš dalies, semantinio VT modelio teisingumo kontrolė.

4.6. VT modelių tikrinimo funkcionalumo įvertinimas

Yra dvi diagramų teisingumo kontrolės priemonės.

1. Sintaksinio teisingumo užtikrinimas: drausti jungti klaidingus ryšius tarp elementų. Šis būdas yra visada aktyvus.
2. Semantinio teisingumo užtikrinimas: drausti neleistinus specialiųjų Roso notacijos elementų naudojimo atvejus. Šie atvejai apibrėžti 2 šio darbo priede. Norint aktyvuoti šį tikrinimą reikia pažymėti varnelę MagicDraw UML meniu Analyze grupės Validation punkto Enable Active Validation papunkčiui.

Atliekant veiksmus buvo stebimi sukurtojo įskiepio VT modelių funkcionalumas (2 lentelė).

2 lentelė. Įskiepio funkcionalumo testavimas

Veiksmas	Gautas rezultatas	Ar atitinka projekto reikalavimus
Bandyamas brėžti ryšį iš klasės neturinčios BRule Anchor stereotipo.	Užvedus pele virš klasės simbolio, jis buvo apibrėžtas raudonai ir neleidžiama jo parinkti ryšio ištaka	Taip
Bandyamas brėžti ryšį į klasės elementą neturinčio Correspondent stereotipo	Užvedus pele virš klasės simbolio, jis buvo apibrėžtas raudonai ir neleidžiama jo parinkti ryšio adresatu	Taip
Bandyamas brėžti liniją iš asociacijos	Užvedus pele virš asociacijos simbolio, jis buvo apibrėžtas raudonai ir neleidžiama jo parinkti linijos ištaka	Taip
Bandyamas brėžti liniją į specialųjį simbolį	Užvedus pele virš specialiojo simbolio, jis buvo apibrėžtas raudonai ir neleidžiama jo parinkti ryšio adresatu	Taip
Specialusis simbolis (Neigimas) kaip interpretatorius sujungiamas ryšiu su sąlygos simboliu, kurio tipas X (Privaloma).	Parodoma klaida, kad specialusis simbolis (Neigimas) panaudotas netinkamai.	Taip
Specialusis simbolis (Visi) kaip apribojimas sujungiamas ryšiu su sąlygos simboliu, kurio tipas LIM (Apribota).	Parodoma klaida, kad specialusis simbolis (Visi) panaudotas netinkamai.	Taip

4.7. Sukurtojo įskiepio įvertinimas

Vertinimas „+“ – yra, „-“ – nėra arba trijų balų sistema, kur: 1 – blogas, 2 - vidutinis, 3 – geras.

3 lentelė. Kriterijų vertinimas ir palyginimas.

Kriterijus	MagicDraw UML įskiepio taisyklėms modeliuoti vertinimas	MS Visio pagrindu sukurto prototipo taisyklėms modeliuoti vertinimas
Patogumas braižyti simbolius	3	2
Simbolių pasirinkimo patogumas	3	3
Užkrautos taisyklės simbolių išdėstymas	3	1
Elemento aprašymo panaudojimas ir susiejimas	+	-
Susiejimas su UML duomenų modeliu	+	-
Specialiųjų simbolių panaudojimo tikrinimas	+	-
Nepabaigtos modeliuoti taisyklės išsaugojimo galimybė	+	-
Užkrautos taisyklės simbolių išdėstymas identiškas išsaugotai	+	-
Taisyklės pakartotinas panaudojimas	+	+

Vertinant VT modeliavimui pagal Roso metodą reikalingas laiko sąnaudas, reikia pastebėti, jog toks taisyklių specifikavimo būdas reikalauja sąlyginai didelių laiko resursų. Todėl galima teigti, jog sukurtuoju įskiepiu tikslinga modeliuoti tuos apribojimus, kurie sistemos projektavimo metu, o dar svarbiau – jau eksploatuojant sukurtą sistemą – turės tendenciją keistis. Akivaizdu, jog būtent šie apribojimai lengviausiai apdorojami veiklos taisyklių pavidalu, o ne pateikiant juos keleto atskirų UML modelių sudėtyje.

5. Išvados

1. Galima teigti, kad bandymų, sukurti visaverčius VT koncepciją palaikančius CASE įrankius, yra, tačiau aptikti įrankiai palaiko tik konkretų VT tipų poaibį. Visavertę grafinę notaciją palaikančių komercinių įrankių aptikta nebuvo.
2. Roso metodas apima plačią nestruktūrinių taisyklių tipų aibę, tačiau dėl savo sąlyginio sudėtingumo jis labiausiai tinka IS projektavimo stadijoje apibrėžiant tik tas veiklos taisykles, kurias realizuotoje sistemoje bus siekiama valdyti centralizuotai, taisyklių procesoriaus pagalba, t.y. kurias reikės dažnai ir operatyviai keisti.
3. Rosas neapibrėžia metodo metamodelio, taip pat nekonkretizuojamas ir duomenų modelio tipas – tai apsunkina metodo taikymą. Tuo tarpu KTU Informacijos sistemų katedroje sukurtas metamodelis yra autonomiškas ir nenumato UML modelių panaudojimo specifikuojant taisykles galimybes.
4. Papildant UML metamodelį, Roso VT buvo susietos su konkrečiu duomenų modeliu (UML klasių diagrama), o tai sukuria prielaidas Roso metodą realizuoti UML kalba grindžiamuose CASE įrankiuose.
5. Atsižvelgiant į technines MagicDraw UML charakteristikas, nuspręsta elementus VT diagramos braižymui sukurti naudojant stereotipus. MagicDraw UML įskiepis yra naudojamas užkrauti profilį su sukurtais stereotipizuotais elementais, jis praplečia klasių metamodelį.
6. Sukurtojo MagicDraw UML įskiepio tyrimai parodė, jog Roso metodas IS projektavimo metu gali būti taikomas kartu su UML. Taip pat nustatyta, jog sukurtas įskiepis pažangesnis funkcinė prasme už esamą KTU sukurtą sprendimą MS Visio aplinkoje.

6. Literatūra

- [1] Business Rules Group. Veiklos taisyklių manifestas. Taisyklių nepriklausomumo principai, ver. 2.0. [interaktyvus] 2003, [žiūrėta 2007-11-18]. Prieiga per internetą: <[http://www.businessrulesgroup.org/brmanifesto/BRManifestLithuanian\(v1.0\).pdf](http://www.businessrulesgroup.org/brmanifesto/BRManifestLithuanian(v1.0).pdf)>.
- [2] Business Rules Group. *Defining Business Rules ~ What Are They Really? (3rd edition)*. [interaktyvus]. 2000, [žiūrėta 2007-10-15]. Prieiga per internetą: <<http://www.BusinesRulesGroup.org>>.
- [3] Butleris, R.; Kapočius, K. (2002). The Business Rules Repository for Information Systems Design. *The 6th East-European Conference ADBIS'2002*: konferencijos pranešimų medžiaga. Bratislava, 2002, p. 64-77.
- [4] Butleris, R., Kapočius, K. Struktūrizuotų veiklos taisyklių saugyklos architektūra. VŠĮ Vilniaus universiteto leidykla [interaktyvus] 2001, [žiūrėta 2007-10-09]. Prieiga per internetą: <<http://www.leidykla.vu.lt/inetleid/inf-mok/17/str6.html>>.
- [5] Haggerty, N. Modeling Business Rules Using the UML and CASE. [interaktyvus] 2005, [žiūrėta 2007-12-09]. Prieiga per internetą: <http://www.brcommunity.com/b016.php> >.
- [6] Leap Systems Company. Modeling Business Rules. [interaktyvus] 2006, [žiūrėta 2007-11-17]. Prieiga per internetą: <<http://www.leapse.com/models.htm>>.
- [7] Kapočius K. Veiklos taisyklių struktūrizavimo modeliai ir jų taikymas kuriant informacijos sistemas: daktaro disertacija. KTU informatikos fakultetas. [Kaunas], 2006. 198 p.
- [8] Morgan, T. (2002). *Business Rules and Information Systems*. New York, 2002. 384 p. ISBN 0201-743-91-4.
- [9] No Magic, Inc. MagicDraw Open API. [interaktyvus] 2007, [žiūrėta 2007-11-08]. Prieiga per internetą: <<http://www.magicdraw.com/files/manuals/15.0>>.
- [10] Object Management Group, Inc. OMG Unified Modeling Language, Superstructure, V2.1.2. [interaktyvus] 2007, [žiūrėta 2008-03-16]. Prieiga per internetą: <<http://www.omg.org/spec/UML/2.1.2/Superstructure/>>
- [11] Ross, R. G. *Principles of the Business Rules Approach*. Boston, 2003. 352 p. ISBN 0201-788-93-4.
- [12] Ross, R. G. *The Business Rule Book: Classifying, Defining and Modeling Rules*. Houston, 1997. 400 p. ISBN 0941-049-03-5.

- [13] Pender, T. (2003). *UML Bible*. Indianapolis, 2003. 940 p. ISBN 0764-526-04-9.
- [14] Pilone, D., Pitman, N. *UML 2.0 in a Nutshell*. Sebastopol, 2005. 234 p. ISBN 0596-007-95-7
- [15] Yasu Technologies Company. Business Rules Life Cycle. [interaktyvus] 2007, [žiūrėta 2007-11-21]. Prieiga per internetą: <<http://www.yasutech.com/qrdn/brlc.htm>>.
- [16] von Halle, B. *Business rules applied: building better systems using the business rules approach*. New York, 2001. 592 p. ISBN 978-0471-412-93-9.

7. MagicDraw UML plugin development and implementation for business rules modeling

Summary

Information system (IS) modeling is popular and constantly developing. However there is a wantage for solutions of realization of computer engineering activities in the business rules based methods. One of the most prominent of this kind of methods is the Ross method, allowing the creation of formal business rules graphic models.









This work specifies and implements business rules modeling tool, which use Ross method, into computer engineering (CASE) tool MagicDraw UML as plugin.










Ross method of its relative complexity is the most suitable for the design phase of IS to define only the operating rules, which realized the system will aim to manage a centralized by business rules engine.










Ross business rules was related to a specific data model (UML class diagram), which creates the preconditions for a method to realize Ross UML-based languages CASE tools. In view of the technical characteristics of MagicDraw UML, decided elements used for diagram create by using stereotypes. MagicDraw UML plugin is used to load a profile with created stereotypes for elements and extended UML metamodel.










8. Priedai






Priedas 1. Specialieji elementai naudojami veiklos taisyklėse (parengta pagal (Ross, 1997)).

Specialus simbolis	Reikšmė
Pagrindas	
 <<stereotype>> Basis T	Tiesa. Testavimo pagrindas - teigimas
 <<stereotype>> Basis F	Netiesa. Testavimo pagrindas - neigiamas
 <<stereotype>> Basis U	Be reikšmės. Pagrindas - be reikšmės, "išjungta"
 <<stereotype>> Basis TF	Tiesa arba Netiesa. Logine kombinacija
 <<stereotype>> Basis TU	Tiesa arba Be reikšmės. Logine kombinacija
 <<stereotype>> Basis FU	Netiesa arba Be reikšmės. Logine kombinacija
 <<stereotype>> Basis TFU	Tiesa arba Netiesa arba Be reikšmės. Logine kombinacija
Aktyvatorius	
 <<stereotype>> Activators C	Sukūrimas.

Specialus simbolis	Reikšmė
 <<stereotype>> Activators M	Modifikavimas
 <<stereotype>> Activators D	Trynimas
 <<stereotype>> Activators UR	Gražinimas
 <<stereotype>> Activators CM	Sukūrimas arba Modifikavimas
 <<stereotype>> Activators MD	Modifikavimas arba Trynimas
 <<stereotype>> Activators CD	Sukūrimas arba Trynimas
 <<stereotype>> Activators CMD	Sukūrimas arba Modifikavimas arba Trynimas
Iškvietimo reikšmės	
 <<stereotype>> Invocation Values III	Neiškviesta
 <<stereotype>> Invocation Values CI	Šiuo metu iškviesta

Specialus simbolis	Reikšmė
 <<stereotype>> Invocation Values IS	Iškviesta sėkmingai
 <<stereotype>> Invocation Values IU	Iškviesta nesėkmingai
 <<stereotype>> Invocation Values III-IS-IU	Neiškviesta arba Iškviesta sėkmingai arba Iškviesta nesėkmingai
 <<stereotype>> Invocation Values IS-IU	Iškviesta sėkmingai arba Iškviesta nesėkmingai
 <<stereotype>> Invocation Values CI-IU	Šiuo metu iškviesta arba Iškviesta nesėkmingai
Specialieji interpretatoriai	
 <<stereotype>> Spec Interpreters II	Neigimas. Taisyklės tipą traktuoti priešingai
 <<stereotype>> Spec Interpreters TY	Lygio pakėlimas. Taisyklė galioja bazės tipui, o ne egzemplioriams.
 <<stereotype>> Spec Interpreters ABZ	Atrinkimas – taikomas bazei. Pagal nutylėjimą.
 <<stereotype>> Spec Interpreters AKR	Atrinkimas – taikomas korespondentui. Įeigos taisyklės bazės egzempliorių pritaikymas.

Specialus simbolis	Reikšmė
 <<stereotype>> Spec Interpreters ABK	Atrinkimas – taikomas abiem. Įeigos taisyklės bazės egzempliorių pritaikymas.
 <<stereotype>> Spec Interpreters IM	Inicijavimas – tuoj pat. Testuoti tuoj pat po reikiamo įvykio.
 <<stereotype>> Spec Interpreters ITR	Inicijavimas – tarpinis. Testuoti kai programa nurodo.
 <<stereotype>> Spec Interpreters CV	Įgalintas – konvergavimas. Leisti priimti taisykle, net jeigu pažeista.
 <<stereotype>> Spec Interpreters SG	Įgalintas – perspėjantis. Leisti pažeistą taisykle, prieš tai paklausus.
 <<stereotype>> Spec Interpreters SL	Įgalintas – miegantis. Niekada nepritaikyti taisyklės, bet išlaikyti tolesnės nuorodoms.
Specialieji apribojimai	
 <<stereotype>> Spec Qualifiers ANY	Bet kuris sąryšis. Sulaužo sąsajas tarp bazės ir korespondento.
 <<stereotype>> Spec Qualifiers ALL	Visi korespondentai. Nuo bazės egzemplioriaus nukreipia į visus korespondentų egzempliorius.
 <<stereotype>> Spec Qualifiers REK	Rekursinis. Rekursinės nuorodos į korespondentų egzempliorius.

Specialus simbolis	Reikšmė
 <<stereotype>> Spec Qualifiers mL	Numeratorius – ne daugiau. Nurodo didžiausią leistiną dydį.
 <<stereotype>> Spec Qualifiers mU	Numeratorius – ne mažiau. Nurodo mažiausią leistiną dydį.
 <<stereotype>> Spec Qualifiers mF	Numeratorius – fiksuotas. Nurodo tikslų leistiną dydį.
 <<stereotype>> Spec Qualifiers n	Numeratorius – skaičius arba atstumas.
 <<stereotype>> Spec Qualifiers LI	Loginis identifikatorius. Nurodo seką, eiliškumą ir pan. pagal nurodyta reikšmę.

Priedas 2. Taisyklingas specialiųjų elementų taisingo panaudojimo aprašas.

Taisyklingas specialiųjų elementų panaudojimas interpretatoriumi.

Taisyklės tipų grupė	Taisyklės tipas	Neigimas	Pagrindas	Iškvietimo reikšmės	Lygio pakėlimas	Atrinkimas	Aktyvatorius	Inicijavimas	Igalintas
Egzempliorių patvirtinimo	X	-	Taip	Taip	Taip	Taip	-	Taip	Taip
	LIM	-	Taip	Taip	Taip	Taip	-	Taip	Taip
Tipų patvirtinimo	M	Taip	Taip	Taip	Taip	-	-	Taip	Taip
	ME	-	Taip	Taip	Taip	-	-	Taip	Taip
	MI	-	Taip	Taip	Taip	-	-	Taip	Taip
	MP	-	Taip	Taip	Taip	-	-	Taip	Taip
Pozicijos patvirtinimo	POS	Taip	Taip	Taip	-	-	Taip	Taip	Taip
	LOW	Taip	Taip	Taip	-	-	Taip	Taip	Taip
	HIGH	Taip	Taip	Taip	-	-	Taip	Taip	Taip
	CHRO	Taip	Taip	Taip	-	-	Taip	Taip	Taip
	OLD	Taip	Taip	Taip	-	-	Taip	Taip	Taip
	NEW	Taip	Taip	Taip	-	-	Taip	Taip	Taip
Funkcinio patvirtinimo	FUNC	Taip	Taip	-	-	-	-	Taip	Taip
	UQ	Taip	Taip	-	-	-	-	Taip	Taip
	FLUC	Taip	Taip	-	-	-	-	Taip	Taip
	AS	Taip	Taip	-	-	-	-	Taip	Taip
	DE	Taip	Taip	-	-	-	-	Taip	Taip
	NRE	-	Taip	-	-	-	-	Taip	Taip
Palyginamojo įvertinimo	EQ	Taip	Taip	-	-	-	-	Taip	Taip
	NE	Taip	Taip	-	-	-	-	Taip	Taip
	GT	Taip	Taip	-	-	-	-	Taip	Taip
	GE	Taip	Taip	-	-	-	-	Taip	Taip
	LT	Taip	Taip	-	-	-	-	Taip	Taip
	LE	Taip	Taip	-	-	-	-	Taip	Taip
Matematinio įvertinimo	CALC	-	Taip	-	Taip	-	-	Taip	Taip
	SUM	-	Taip	-	Taip	-	-	Taip	Taip
	SUB	-	Taip	-	Taip	-	-	Taip	Taip
Projekcijos valdikliai	EA	-	Taip	Taip	Taip	Taip	-	Taip	Taip
	COP	-	Taip	Taip	Taip	-	-	Taip	Taip
	EX	-	Taip	Taip	Taip	-	-	Taip	Taip

Taisyklingas specialiųjų elementų panaudojimas apribojimu.

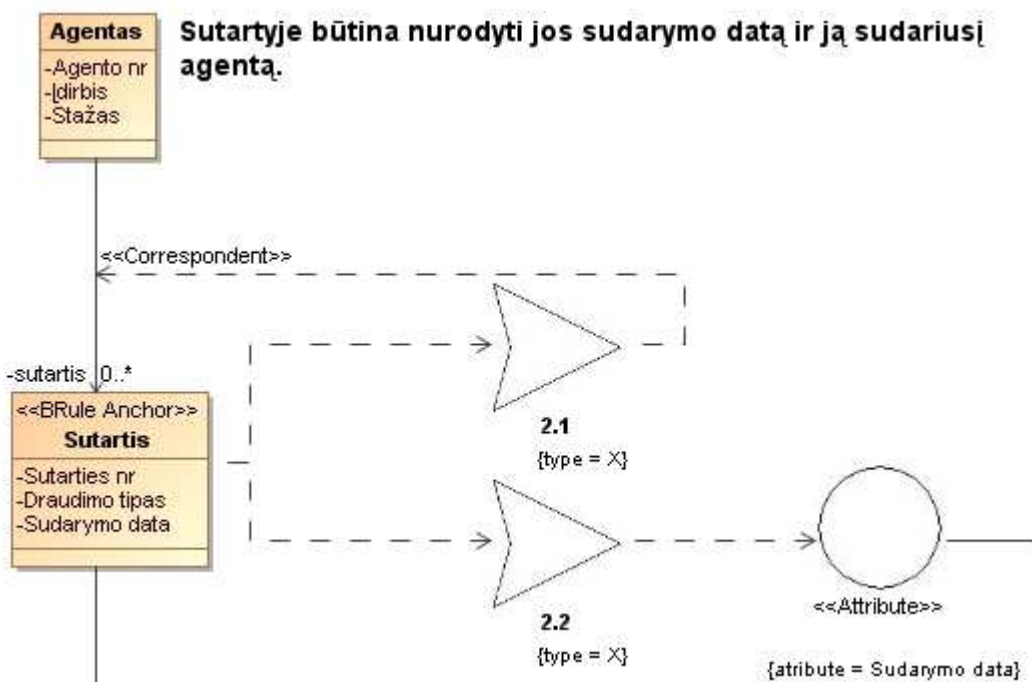
Taisyklės tipų grupė	Taisyklės tipas	Pagrindas	Lygio pakėlimas	Bet kuris	Visi	Rekursija	Numeratorius	Loginis identifikatorius	Aktyvatorius	Iškvietymo reikšmės
Egzempliorių patvirtinimo	X	Taip	-	Taip	Taip	Taip	-	-	-	Taip
	LIM	Taip	-	Taip	-	Taip	Taip	-	-	Taip
Tipa patvirtinimo	M	Taip	-	Taip	-	Taip	Taip	-	-	Taip
	ME	Taip	-	Taip	-	Taip	-	-	-	Taip
	MI	Taip	-	Taip	-	Taip	-	-	-	Taip
	MP	Taip	-	Taip	-	Taip	-	-	-	Taip
Pozicijos patvirtinimo	POS	-	Taip	Taip	-	Taip	-	Taip	Taip	-
	LOW	-	Taip	Taip	-	Taip	-	Taip	Taip	-
	HIGH	-	Taip	Taip	-	Taip	-	Taip	Taip	-
	CHRO	-	Taip	Taip	-	Taip	-	Taip	Taip	-
	OLD	-	Taip	Taip	-	Taip	-	Taip	Taip	-
	NEW	-	Taip	Taip	-	Taip	-	Taip	Taip	-
Funkcinio patvirtinimo	FUNC	-	-	-	-	-	Taip	-	-	-
	UQ	-	-	-	-	-	Taip	-	-	-
	FLUC	-	-	-	-	-	Taip	-	-	-
	AS	-	-	-	-	-	Taip	-	-	-
	DE	-	-	-	-	-	Taip	-	-	-
	NRE	Taip	-	-	-	-	Taip	-	-	-
Palyginamojo įvertinimo	EQ	-	-	Taip	Taip	Taip	Taip	-	-	-
	NE	-	-	Taip	Taip	Taip	Taip	-	-	-
	GT	-	-	Taip	Taip	-	Taip	-	-	-
	GE	-	-	Taip	Taip	-	Taip	-	-	-
	LT	-	-	Taip	Taip	-	Taip	-	-	-
	LE	-	-	Taip	Taip	-	Taip	-	-	-
Matematinio įvertinimo	CALC	-	-	Taip	-	Taip	-	Taip	-	-
	SUM	-	-	Taip	-	Taip	-	Taip	-	-
	SUB	-	-	Taip	-	Taip	-	Taip	-	-
Projekcijos valdikliai	EA	Taip	-	Taip	-	Taip	Taip	-	-	-
	COP	-	-	Taip	-	Taip	Taip	-	-	-
	EX	-	-	Taip	-	Taip	Taip	-	-	Taip

Priedas 3. Roso metodo veiklos taisyklių klasifikacija.

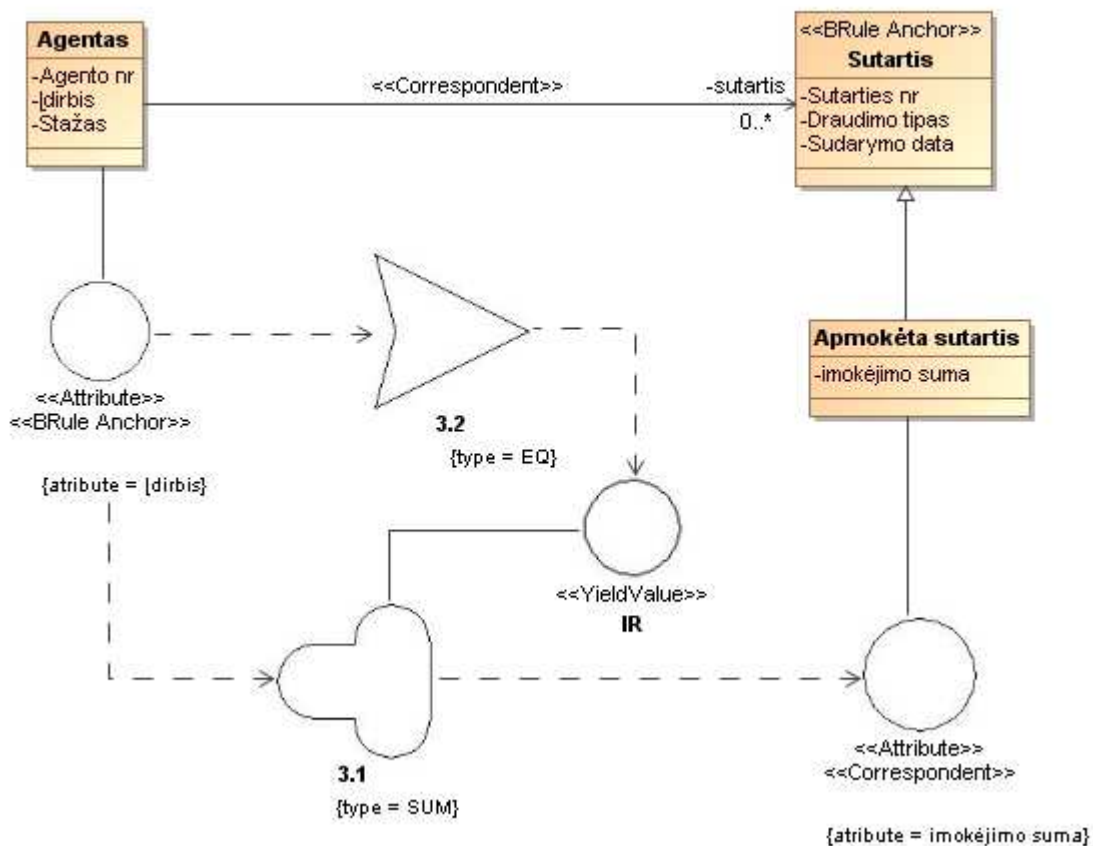
Detali Roso pasiūlytos VT klasifikacijos specifikacija pateikiama [Rosas, 1997]. Žemiau pateikiamas tipų sąrašas, sudarytas remiantis Roso metodo VT klasifikacijos vertimu į lietuvių kalbą, pateiktu (Kapočius, 2006).

Taisyklės tipų grupė	Taisyklės tipo santrumpa	Tipas
Egzempliorių patvirtinimo	X	Privaloma
	LIM	Apribota
Tipo patvirtinimo	M	Bendra
	ME	Bendrai-atiskirianti
	MI	Bendrai-susiejanti
	MP	Bendrai-uždraudžianti
Pozicijos patvirtinimo	POS	Pozicinė
	LOW	Žemiausia
	HIGH	Aukščiausia
	CHRO	Chronologinė
	OLD	Seniausia
	NEW	Naujausia
Funkcinio patvirtinimo	FUNC	Funkcinė
	UQ	Unikali
	FLUC	Svyruojanti
	AS	Auganti
	DE	Krentanti
	NRE	Neatnaujinama
Palyginamojo įvertinimo	EQ	Lygu
	NE	Nelygu
	GT	Daugiau-už
	GE	Daugiau-arba-lygu
	LT	Mažiau-už
	LE	Mažiau-arba-lygu
Matematinio įvertinimo	CALC	Apskaičiuota
	SUM, SUB, MULT, ect.	Sudėtis, atimtis, daugyba ir t.t.
Projekcijos valdikliai	EA	Įgalinta
	COP	Nukopijuota
	EX	Įvykdyta

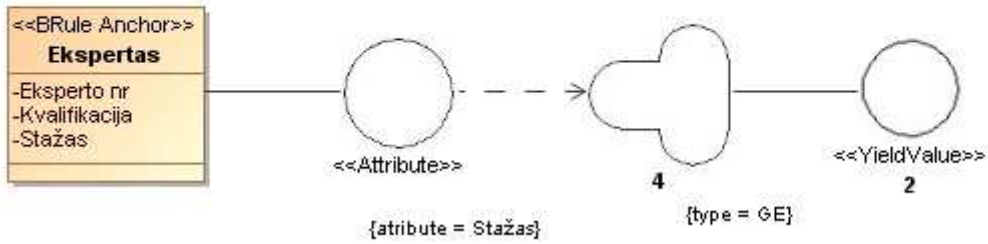
Priedas 4. Veiklos taisyklės sumodeliuotos sukurtuoju MagicDraw UML įskiepiu pagal šio darbo 4.3 skyriuje suformuluotas taisykles.



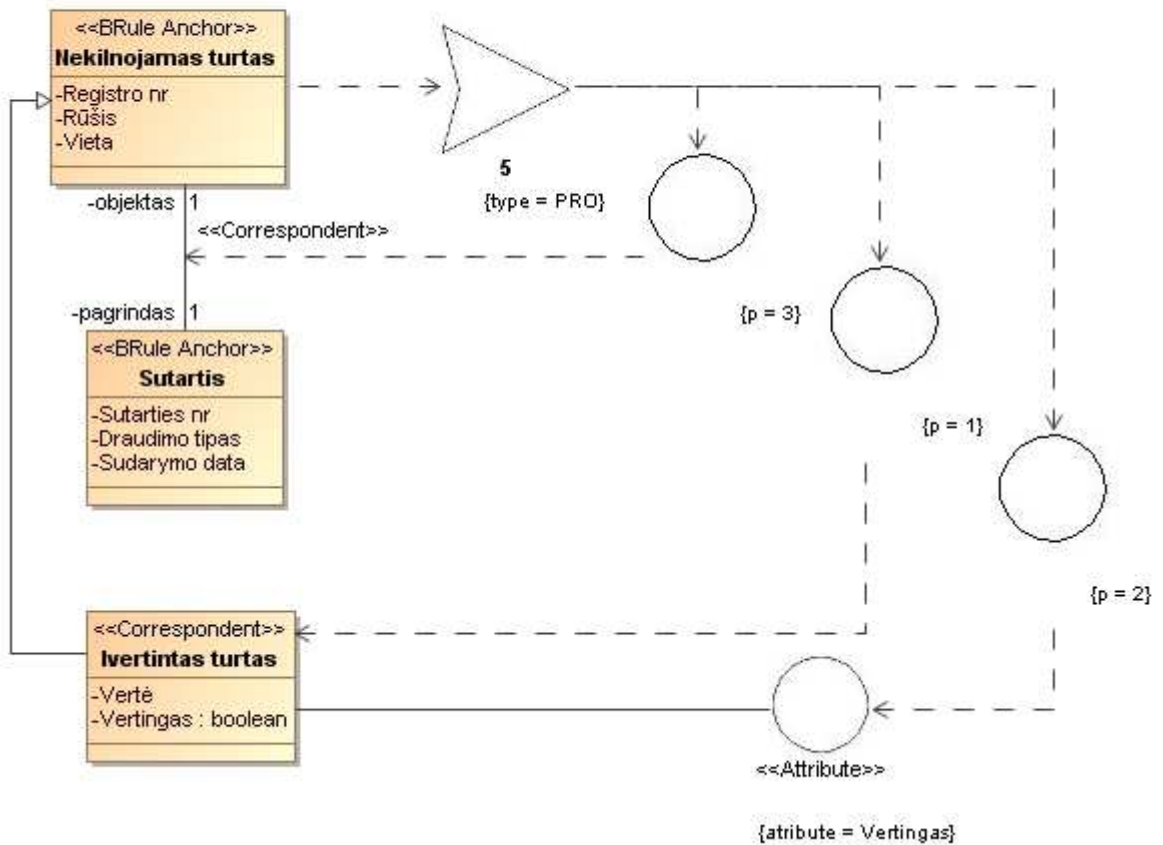
Agento įdirbis privalo būti lygus apmokėtų sutarčių, sudarytų šio agento, bendrai įmokėjimo sumų sumai.



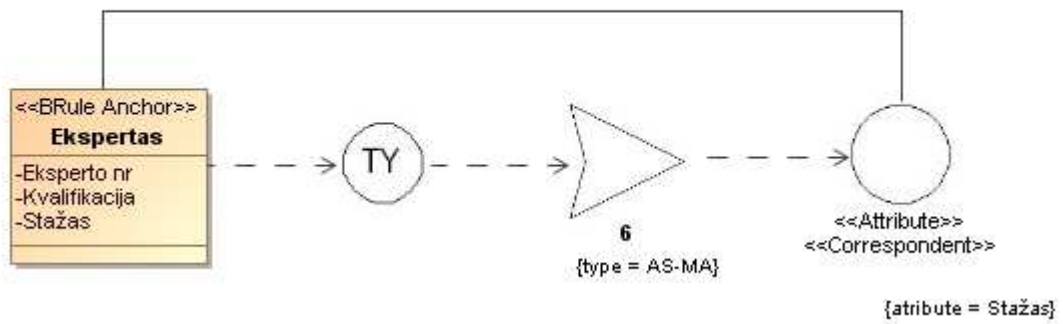
Eksperto stažas privalo būti didesnis arba lygus 2 metams



Jeigu nekilnojamasis turtas yra įvertintas, tačiau nėra klasifikuojamas kaip pakankamai vertingas ir pagal jį nesudaryta sutartis, tai jis privalo būti klasifikuojamas kaip pakankamai vertingas prieš tai, kai pagal jį galės būti sudaryta sutartis.



Įvedant naują arba modifikuojant esamą visų ekspertų stažą, jis privalo didėti.



Nekilnojamasis turtas negali būti įtrauktas į daugiau nei 5 sutartis

